JOSÉ AUGUSTO FABRI

Uma Proposta de Modelo para a Criação e a Organização de Processos de
Produção em um Contexto de Fábrica de Software

São Paulo
2007

JOSÉ AUGUSTO FABRI

Uma Proposta de Modelo para a Criação e a Organização de Processos de
Produção em um Contexto de Fábrica de Software

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção
do título de Doutor em Engenharia

São Paulo
2007

JOSÉ AUGUSTO FABRI

Uma Proposta de Modelo para a Criação e a Organização de Processos de
Produção em um Contexto de Fábrica de Software

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção
do título de Doutor em Engenharia

Área de Concentração:
Engenharia da Produção

Orientador:
Prof. Dr. Marcelo Schneck de Paula Pessoa

São Paulo
2007

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 24 de maio de 2007

FICHA CATALOGRÁFICA

Fabri, José Augusto

Uma proposta de modelo para a criação e a organização de processos de produção em um contexto de fábrica de software / J.A. Fabri. -- Ed.Rev. -- São Paulo, 2007.

268 p. + anexos

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Engenharia de software 2.Processo de software 3.Produtividade de software I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

DEDICATÓRIA

Dedico este trabalho
a Marília Gabriela de Souza Fabri
e a Heloísa de Souza Fabri.

AGRADECIMENTOS

Agradeço a DEUS por me conceber o dom da vida.

Ao Prof. Dr. Marcelo Schneck de Paula Pessôa, pela oportunidade, orientação e pelo constante estímulo transmitido durante todo o trabalho.

Ao amigo André Luiz Prezende Trindade pelas discussões e incentivo delineados durante o processo de doutoramento.

As empresas que se propuseram a participar dos estudos de casos apresentados neste trabalho.

A Fundação Educacional do Município de Assis e a Faculdade de Tecnologia de Ourinhos, instituições estas que disponibilizaram toda infra-estrutura para o desenvolvimento deste trabalho.

“Construirei um carro para as grandes massas, feito com os melhores materiais, pelos melhores homens que puderem ser contratados e seguindo os projetos mais simples que a moderna engenharia pode conceber [...] de preço tão baixo que qualquer homem que ganhe um bom salário seja capaz de possuir – e desfrutar com sua família a bênção das horas de prazer nos grandes espaços abertos da natureza”.

Henry Ford (início de sua carreira como produtor de carros)

RESUMO

Este trabalho tem como objetivo propor um modelo para a criação e organização de um processo fabril de produção de software. Para atingir este objetivo foram mapeadas 11 empresas de produção de software com características fabris, 6 brasileiras (o autor deste trabalho não possui uma autorização formal para divulgar o nome das empresas) e 5 estrangeiras (as japonesas Hitachi, Toshiba, NEC, Fujitsu e a americana SDC). Salienta-se que os dados utilizados neste trabalho sobre as empresas estrangeiras foram extraídos de CUSUMANO (1991). É importante salientar que todas as fábricas brasileiras que se propuseram a participar do estudo de caso possuem certificação de qualidade em processos comprovada (CMMI e/ou ISO). Após a apresentação dos casos é realizada uma comparação entre os processos fabris brasileiros e estrangeiros. Uma aderência do processo de produção de software mapeados nas empresas ao modelo proposto, também, é desenvolvida no trabalho. Por fim, 01 caso real apresentando o comportamento do modelo proposto na criação de um processo fabril, também, se caracteriza como um dos pontos a ser destacado.

Palavras Chaves: Fábrica de software. Processo de software. Meta-processo de software. Reuso de Processo.

ABSTRACT

This work has as objective to propose a model to create and to organize a production process with factory software characteristic. To reach this objective 11 software production companies were mapped, 6 Brazilian (the author of this work doesn't possess an authorization to publish the name of the companies) and 5 foreigners (Hitachi, Toshiba, NEC, Fujitsu and SDC). The data used in this work, on the foreign companies were extracted of CUSUMANO (1991). All the Brazilian factories that participate of this case study possess quality certification in processes (CMMI and/or ISO). After the presentation of the cases a comparison between the Brazilian factories and the foreigners' factories is developed. An adherence of the software production process mapped in the companies to the proposed model, also, is showed in the work. A real case presenting the behavior of the model proposed in the creation software production process, also, is characterized in the text (12 cases in the total).

Keywords: Software Factory. Software process. Software meta-process. Process reuse

LISTA DE ILUSTRAÇÕES

Figura 1.1	- Classificação, em %, das empresas em Micro, Pequena, Média e Grande.....	024
Figura 1.2	- Distribuição geográfica do mercado global de <i>outsourcing</i> de serviços de TI.....	031
Figura 1.3	- Evolução do mercado global de <i>outsourcing</i> por segmento de escala de serviços de TI.....	032
Figura 1.4	- Tamanho dos principais mercados de software e serviços de TI – 2003.....	033
Figura 2.1	- Modelo evolucionário (orientado a qualidade)	042
Figura 2.2	- Modelo orientado à qualidade (norma ISO 9000 e modelo CMM).....	044
Figura 2.3	- A associação proposta entre as visões gerencial e processual.....	046
Figura 2.4	- Integração entre o modelo proposto por LI et. al. (2001) com o modelo composto de categorização por níveis	047
Figura 2.5	- Visão embrionária de uma fábrica de software com processo segmentado.....	049
Figura 2.6	- Fábrica experimental (evolução da visão embrionária).....	050
Figura 2.7	- Modelo classificatório proposto para definição do escopo de fornecimento de uma fábrica de software.....	051
Figura 2.8	- Modelo de fábrica de programas.....	052
Figura 2.9	- Linha de processo instanciada de uma fábrica de programa.....	056
Figura 2.10	- Fábrica de projeto não dedicada e generalista.....	058
Figura 2.11	- Fábrica de projeto dedicada e não generalista.....	060
Figura 3.1	- Visão do aspecto produtivo de software.....	074
Figura 3.2	- Modelo Eureka.....	075
Figura 3.3	- Barramento de produção software.....	075
Figura 3.4	- Visão embrionária do modelo organizacional para fábrica de software do ELABSOFT.....	077
Figura 3.5	- Evolução da visão embrionária do modelo organizacional para fábrica de software do ELABSOFT.....	078
Figura 3.6	- Processo fabril ELABSOFT.....	081
Figura 3.7	- Extensão do modelo classificatório de Fernandes e Teixeira.....	085
Figura 3.8	- Modelo dinâmico do processo fabril em uma fábrica de software.....	087
Figura 3.9	- Processo gestor: o gerenciamento do processo fabril.....	091

Figura 4.1 -	Universalidade de processo de software.....	101
Figura 4.2 -	Visão embrionária de um processo.....	102
Figura 4.3 -	Modelo caótico.....	103
Figura 4.4 -	Modelo cascata.....	104
Figura 4.5 -	Modelo incremental.....	105
Figura 4.6 -	Modelo espiral.....	106
Figura 4.7 -	Modelo prototipagem.....	107
Figura 4.8 -	Composição de um processo de produção de software (segundo PAULA FILHO 2003)	109
Figura 4.9 -	Composição de um processo de produção de software (segundo DERNIAME et. al. (1999))	110
Figura 4.10 -	Composição de um processo de produção de software (visão deste trabalho)	111
Figura 4.11 -	Cenário geral para a reutilização de processos de software.....	116
Figura 4.12 -	<i>Framework</i> modular de reuso.....	117
Figura 4.13 -	Relacionamento das atividades proposta no modelo 3C.....	128
Figura 4.14 -	Digrama de seqüência ilustrando a execução de um processo de software.....	134
Figura 4.15 -	Modelagem estática de um processo.....	135
Figura 4.16 -	Estrutura arquitetônica do modelo proposto por Henninger.....	143
Figura 4.17 -	Modelo arquitetônico para reutilização de processos.....	144
Figura 5.1 -	Visões utilizadas na concepção do modelo.....	149
Figura 5.2 -	Visão embrionária do modelo.....	149
Figura 5.3 -	ICOM representando o padrão para modelagem de processo.....	153
Figura 5.4 -	Níveis de abstração de um componente em relação aos domínios de aplicação.....	156
Figura 5.5 -	Visão estendida do modelo.....	162
Figura 6.1 -	Processo de evolução científica.....	169
Figura 6.2 -	O ato de pesquisar.....	170
Figura 6.3 -	Modelo utilizado na composição do roteiro para o desenvolvimento estudo de caso (Etapa 1).....	178
Figura 6.4 -	Forma de descrição dos fatos mapeados nos estudos de casos.....	181
Figura 8.1 -	Estruturação física das fábricas brasileiras.....	200
Figura 8.2 -	Modelo dinâmico do processo fabril em uma fábrica (ELABSOFT), incorporando as atividades de levantamento de requisitos,	

	equalização, gestão de configuração e gestão da qualidade e produto.....	202
Figura 9.1 -	Modelagem do processo de produção de software da <u>EMPRESA A</u>	209
Figura 9.2 -	Modelagem das atividades de gestão de projeto, configuração e da qualidade.....	210
Figura 9.3 -	Modelagem das tarefas da atividade de equalização.....	213
Figura 9.4 -	Estrutura do modelo – evolução 1.....	217
Figura 10.1 -	Modelo de preservação do conhecimento.....	223
Figura 10.2 -	Atividades do processo de produção da fábrica de software da FATEC-OU.....	224
Figura 10.3 -	Modelagem das tarefas da atividade de codificação.....	227
Figura 10.4 -	Organização da máquina de processo.....	227
Figura 10.5 -	História de vida do componente.....	230
Figura 10.6 -	Cronograma e definição dos recursos para implantação do processo de produção de software.....	231
Figura 10.7 -	Diagrama de seqüência gerado na atividade de projeto de software	232
Figura 10.8 -	Ordem de serviço para codificação.....	234
Figura 10.9 -	Ordem de serviço para teste.....	234
Figura 10.10 -	Acesso real de um desenvolvedor a uma ordem de serviço.....	241
Figura 10.11 -	Estrutura do Modelo – Evolução 2	245
Figura A.1 -	Estrutura organizacional da fábrica de software da <u>EMPRESA A</u>	271
Figura A.2 -	Modelo de Produção da <u>EMPRESA A</u>	272
Figura A.3 -	A atividade de equalização – <u>EMPRESA A</u>	275
Figura A.4 -	Instanciando código a partir da plataforma de desenvolvimento – <u>EMPRESA A</u>	277
Figura A.5 -	Procedimento de teste desenvolvido pela fábrica de software da <u>EMPRESA A</u>	279
Figura A.6 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA A</u> no modelo de categorização por níveis.....	283
Figura A.7 -	Diagrama detalhando a atividade de gestão de projeto – <u>EMPRESA A</u>	286
Figura A.8 -	Esquemático definido para gestão de configuração (<u>EMPRESA A</u>)...	288
Figura A.9 -	Estrutura da máquina de processo da fábrica de software da <u>EMPRESA A</u>	290
Figura A.10 -	Estrutura organizacional da fábrica de software da <u>EMPRESA B</u>	295

Figura A.11 -	Modelo de produção da <u>EMPRESA B</u>	296
Figura A.12 -	Atividades da fábrica de software da <u>EMPRESA B</u>	298
Figura A.13 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA B</u> no modelo de categorização por níveis.....	302
Figura A.14 -	Atividade de planejamento da fábrica de software (<u>EMPRESA B</u>).....	304
Figura A.15 -	Modelo de produção da <u>EMPRESA C</u>	309
Figura A.16 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA C</u> no modelo de categorização por níveis.....	314
Figura A.17 -	Atividade de gestão de projetos da fábrica de código e de teste da <u>EMPRESA C</u>	316
Figura A.18 -	Estrutura organizacional da <u>EMPRESA D</u>	319
Figura A.19 -	Configuração do ciclo curto de produção da unidade de produção de software, caracterizada como fábrica (<u>EMPRESA D</u>).....	321
Figura A.20 -	Configuração do ciclo médio de produção da unidade de produção de software, caracterizada como fábrica (<u>EMPRESA D</u>)	322
Figura A.21 -	Exemplo de um artefato utilizado na atividade de teste pela fábrica de software (<u>EMPRESA D</u>).....	325
Figura A.22 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA D</u> no modelo de categorização por níveis.....	328
Figura A.23 -	Configuração da atividade de gestão de projetos (<u>EMPRESA D</u>).....	330
Figura A.24 -	Estrutura organizacional da fábrica de software da <u>EMPRESA E</u>	335
Figura A.25 -	Modelo de produção da <u>EMPRESA E</u>	336
Figura A.26 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA E</u> no modelo de categorização por níveis.....	341
Figura A.27 -	Configuração da atividade de gestão de projetos (<u>EMPRESA E</u>).....	344
Figura A.28 -	Estrutura Organizacional da <u>EMPRESA F</u>	349
Figura A.29 -	O Ciclo de Produção praticado pela <u>EMPRESA F</u>	350
Figura A.30 -	Distribuição dos envolvidos com o processo de produção da <u>EMPRESA F</u> no modelo de categorização por níveis.....	355
Figura B.1 -	Modelo de organização da produção e delimitação dos ciclos da SDC.....	369
Figura B.2 -	Processo de produção de software da Hitachi.....	372
Figura B.3 -	Processo de produção de software da Toshiba.....	374
Figura B.4 -	Relação das atividades do processo de produção de software – Fujitsu.....	378
Figura B.5 -	Rastreabilidade no processo de teste para controle da qualidade	

	(Toshiba).....	384
Figura B.6 -	Relação entre o processo de produção de computadores (fábrica convencional) e o processo de produção de software (fábrica de software) da Toshiba.....	398

OBS: As figuras apresentadas nesta tese são de autoria de seu autor, exceto aquelas em que haja a declaração “Extraída de...”. As idéias expressas por figuras não afeitas a esta exceção são representações das formas como o autor deste documento de tese interpreta fatos, situações ou modelos conceituais representativos de ambientes, contextos, organizações ou processos.

LISTA DE TABELAS

Tabela 1.1 -	Número de empresas com atividades de informática divididas por estado.....	023
Tabela 1.2 -	Áreas relacionadas a TI em que a organização atua além da atividade de software.....	025
Tabela 1.3 -	Certificados ISO 9001 emitidos no mundo (classificação por continente).....	026
Tabela 1.4 -	Certificados ISO 9001 emitidos por países da América do Sul.....	027
Tabela 1.5 -	Números de certificações ISO emitidos por estado da federação.....	027
Tabela 1.6 -	Certificações ISO 9001 para as empresas com atividade na área de informática e conexas.....	028
Tabela 1.7 -	Certificação CMMI em âmbito mundial, em relação ao tamanho das empresas.....	029
Tabela 1.8 -	Certificações CMMI em nível mundial.....	030
Tabela 4.1 -	Comparação entre os processos de produção: culinário e de software.....	099
Tabela 4.2 -	Algumas operações descritas como processo de <i>input-transformação-output</i>	101
Tabela 4.3 -	Dados consolidados de produtividade de uma pessoa.....	136
Tabela 4.4 -	Análises estatísticas sobre a produtividade de uma pessoa.....	136
Tabela 4.5 -	Guia de Reutilização.....	145
Tabela 5.1 -	Visões: das atividades para criação e organização de um processo do modelo 3C de HOLLENBANCH e FRANKES e do META-PROCESSO.....	166
Tabela 6.1 -	Situações relevantes para diferentes estratégias de pesquisa.....	171
Tabela 6.2 -	Lista de casos apresentados neste trabalho.....	174
Tabela 6.3 -	Protocolo utilizado para a execução da etapa 1 do estudo de caso	176
Tabela 6.4 -	Protocolo utilizado para a execução da etapa 2 do estudo de caso	178
Tabela 7.1 -	Relação das ferramentas e atividades do processo nas fábricas de A e F.....	189
Tabela 8.1 -	Presença das atividades do processo de produção de software do modelo ELABSOFT nas empresas pesquisadas.....	201
Tabela 8.2 -	Relação entre as empresas brasileiras e estrangeiras.....	204
Tabela 9.1 -	Legendas e descritores da Figura 9.1.....	210
Tabela 9.2 -	Formalização das características das informações geradas pela	

	máquina de processos.....	215
Tabela 10.1 -	<i>Checklist</i> de teste.....	235
Tabela 10.2 -	Configuração da atividade de equalização durante a EXECUÇÃO do processo.....	239
Tabela A.1 -	Relação das ferramentas e atividades do processo – <u>EMPRESA A</u>	289
Tabela A.2 -	Características das Informações geradas pela máquina de processo da <u>EMPRESA A</u>	291
Tabela A.3 -	Relação das ferramentas e atividades do processo – <u>EMPRESA B</u>	305
Tabela A.4 -	Relação das ferramentas e atividades do processo – <u>EMPRESA C</u>	316
Tabela A.5 -	Descrição das metas e práticas específicas que compõem a modelagem da atividade de teste da Fábrica de Software - <u>EMPRESA D</u>	324
Tabela A.6 -	Relação das ferramentas e atividades do processo – <u>EMPRESA D</u>	331
Tabela A.7 -	Relação das ferramentas e atividades do processo – <u>EMPRESA E</u>	345
Tabela A.8 -	Relação das ferramentas e atividades do processo – <u>EMPRESA F</u>	357
Tabela B.1 -	Especificação das atividades pertinentes ao processo de produção de software da NEC.....	375
Tabela B.2 -	Projetos de software atrasados – Hitachi.....	381
Tabela B.3 -	Média de falha de sistemas apontados pelos clientes – Hitachi.....	381
Tabela B.4 -	Produtividade da fábrica de software da Toshiba.....	383
Tabela B.5 -	Especificação das tarefas de planejamento e de controle da atividade de gestão de projetos da NEC.....	387
Tabela B.6 -	Resultados aferidos após o desenvolvimento do programa de qualidade da NEC.....	388
Tabela B.7 -	Relação entre os critérios de qualidade de software e qualidade dos requisitos – NEC.....	389
Tabela B.8 -	Produtividade do software para o domínio bancário – Fujitsu.....	390
Tabela B.9 -	Defeitos em linhas de código reportados pelos envolvidos com o processo e pelos usuários – Fujitsu.....	391

LISTA DE QUADROS

Quadro 2.1 -	Atributos para uma fábrica de software.....	041
Quadro 4.1 -	Caso de uso descrevendo parte de um processo de software.....	132
Quadro 9.1 -	Apresentação do artefato: documento de rejeite – <u>EMPRESA A</u>	214
Quadro 10.1 -	Apresentação do artefato ordem de serviço – gerada pela máquina de processo – fábrica de software da FATEC-OU.....	228
Quadro 10.2 -	Características do software a ser desenvolvido pela FATEC-OU (Atividade de execução).....	236
Quadro 10.3 -	E-mail estabelecendo a atividade de equalização.....	238

SUMÁRIO

1 Introdução.....	022
1.1 Objetivos e Caracterização do Trabalho.....	035
1.2 Estrutura Metodológica para o Desenvolvimento do Trabalho.....	037
1.3 Composição Estrutural do Trabalho.....	039
2 Fábrica de Software.....	041
2.1 As Fábricas Estrangeiras.....	061
2.1.1 Caracterização das Empresas.....	061
2.1.2 Insumos e Atividades do Processo de Produção de Software.....	063
2.1.3 Gestão Operacional, da Qualidade e de Configuração.....	066
2.1.4 Ferramentas e Biblioteca de Componentes.....	069
2.1.5 Produtos Gerados e Forma de Organização do Processo.....	070
2.2 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo.....	071
3 O Modelo de Fábrica de Software Proposto pelo ELABSOFT.....	073
3.1 A Proposta Efetuada pelo ELABSOFT.....	076
3.2 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo.....	095
4 Formalização dos Conceitos de Meta-Processo e Reuso de Processo de Software.....	097
4.1 Definições sobre Processo de Software.....	098
4.2 Evolução dos Modelos Procedimentais de Produção de Software.....	102
4.3 Meta-processo e Reuso de Processo de Produção de Software.....	111
4.4 Requisitos para a Definição de um Processo.....	118
4.5 Modelos de Reuso de Processo Advindos da Literatura.....	125
4.5.1 – Modelo 3C de HOLLENBANCH e FRAKES.....	125
4.5.2 – Modelo Gertude de SUCCI, BENEDICENTI, PREDONZANI e VERNAZZA.....	129
4.5.3 – Modelo de BECKER, HAMANN E VERLAGE.....	137
4.5.4 – Modelo de HENNINGER.....	142
4.5.5 – O Modelo Arquitetônico de Reutilização de Processos de FIORINI.....	143
4.6 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo.....	147
5 Proposta Preliminar do Modelo para a Criação e a	

Organização de Processos de Produção em um Contexto de Fábrica de Software.....	148
5.1 Atividade de Modelagem.....	152
5.2 Atividade de Instanciação.....	154
5.3 Atividade de Simulação.....	160
5.4 Atividade de Execução.....	160
5.5 Atividade de Avaliação.....	160
5.6 Atividade de Armazenamento.....	161
5.7 Requisitos Utilizados na Visão de Reuso de Processos.....	164
5.8 Atividade de Adaptação (Visão de Reuso de Processos).....	165
5.9 Considerações Finais em Relação ao Modelo Preliminar.....	166
6 Método de Pesquisa utilizado no Desenvolvimento do Trabalho.....	168
7 As Fábricas Brasileiras.....	182
7.1 – Caracterização das Empresas.....	182
7.2 – Insumos e Atividades do Processo de Produção de Software.....	183
7.3 – Gestão Operacional, da Qualidade e de Configuração.....	187
7.4 – Ferramentas e Biblioteca de Componentes.....	188
7.5 – Produtos Gerados e Forma de Organização do Processo.....	189
8 Análise Comparativa entre as Fábricas Brasileiras e Estrangeiras.....	191
9 A Aderência do META-PROCESSO aos Casos Brasileiros, Japoneses e Americano.....	205
9.1 A Aderência dos Casos aos Requisitos do META-PROCESSO.....	205
9.2 A Aderência dos Casos a Atividade de Modelagem do META-PROCESSO.....	207
9.3 A Aderência dos Casos a Atividade de Instanciação do META-PROCESSO.....	211
9.4 A Aderência dos Casos as Bases de Dados do META-PROCESSO.....	215
9.5 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo.....	216
10 Experimento Prático do META-PROCESSO.....	218
10.1 Histórico e Descrição do Ambiente Experimental.....	218
10.2 O Levantamento de Requisitos para Definição do Processo de	

Produção de Software.....	220
10.3 A Atividade de Modelagem.....	224
10.4 A Atividade de Instanciação.....	225
10.4.1 Definindo Tarefas e Ativos do Processo.....	225
10.4.2 Estruturando a Máquina de Processo.....	228
10.5 A Atividade de SIMULAÇÃO.....	231
10.6 A Atividade de EXECUÇÃO.....	236
10.6.1 Execução da Atividade de Equalização.....	237
10.6.2 Execução da Atividade de Codificação.....	240
10.6.3 Execução da Atividade de Teste.....	241
10.7 A Atividade de AVALIAÇÃO.....	242
10.8 – As Atividades de ARMAZENAMENTO, SELEÇÃO e ADAPTAÇÃO..	243
10.9 – Conclusões inferidas a partir das Considerações Efetuadas neste Capítulo.....	244
11 Conclusões e Trabalhos Futuros.....	246
11.1 Contribuições do Trabalho junto ao Pilar Acadêmico.....	246
11.2 Contribuições do Trabalho junto ao Pilar Empresarial.....	249
11.3 Verificação das Proposições e a Resposta da Questão de Pesquisa.....	251
11.4 A Relação do Modelo com a Teoria de Fábrica de Software.....	253
11.5 Trabalhos Futuros.....	255
Referências Bibliográficas.....	256
ANEXO A Relato Detalhado das Fábricas Brasileiras.....	269
A1 A Fábrica de Software da <u>EMPRESA A</u>	269
A.1.1 Caracterização da Empresa.....	269
A.1.2 Insumos do Processo.....	272
A.1.3 Atividades do Processo.....	273
A.1.4 Gestão Operacional.....	283
A.1.5 Armazenamento de Dados.....	288
A.1.6 Produtos Gerados.....	292
A.1.7 Forma de Criação e Organização do Processo.....	293
A.2 A Fábrica de Software da <u>EMPRESA B</u>	293
A.2.1 Caracterização da Empresa.....	293
A.2.2 Insumos do Processo.....	295

A.2.3 Atividades do Processo.....	297
A.2.4 Gestão Operacional.....	303
A.2.5 Armazenamento de Dados.....	305
A.2.6 Produtos Gerados.....	305
A.2.7 Forma de Criação e Organização do Processo.....	306
A.3 A Fábrica de Software da <u>EMPRESA C</u>.....	306
A.3.1 Caracterização da Empresa.....	306
A.3.2 Insumos do Processo.....	308
A.3.3 Atividades do Processo.....	310
A.3.4 Gestão Operacional.....	314
A.3.5 Armazenamento de Dados.....	316
A.3.6 Produtos Gerados.....	317
A.3.7 Forma de Criação e Organização do Processo.....	317
A.4 A Fábrica de Software da <u>EMPRESA D</u>.....	318
A.4.1 Caracterização da Empresa.....	318
A.4.2 Insumos do Processo.....	320
A.4.3 Atividades do Processo.....	322
A.4.4 Gestão Operacional.....	328
A.4.5 Armazenamento de Dados.....	331
A.4.6 Produtos Gerados.....	332
A.4.7 Forma de Criação e Organização do Processo.....	332
A.5 A Fábrica de Software da <u>EMPRESA E</u>.....	333
A.5.1 Caracterização da Empresa.....	333
A.5.2 Insumos do Processo.....	335
A.5.3 Atividades do Processo.....	336
A.5.4 Gestão Operacional.....	342
A.5.5 Armazenamento de Dados.....	345
A.5.6 Produtos Gerados.....	346
A.5.7 Forma de Criação e Organização do Processo.....	346
A.6 A Fábrica de Software da <u>EMPRESA F</u>.....	347
A.6.1 Caracterização da Empresa.....	347
A.6.2 Insumos do Processo.....	349
A.6.3 Atividades do Processo.....	351
A.6.4 Gestão Operacional.....	355
A.6.5 Armazenamento de Dados.....	356
A.6.6 Produtos Gerados.....	357

A.6.7 Forma de Criação e Organização do Processo.....	358
---	-----

ANEXO B Relato Detalhado das Fábricas Japonesas e Americana..... 359

B.1 Caracterização das Empresas.....	359
--------------------------------------	-----

B.1.1 Caracterização da SDC.....	359
----------------------------------	-----

B.1.2 Caracterização da Hitachi.....	361
--------------------------------------	-----

B.1.3 Caracterização da Toshiba.....	362
--------------------------------------	-----

B.1.4 Caracterização da NEC.....	363
----------------------------------	-----

B.1.5 Caracterização da Fujitsu.....	364
--------------------------------------	-----

B.2 O Processo de Produção de Software das Empresas.....	365
--	-----

B.2.1 Processo de Produção da SDC.....	366
--	-----

B.2.2 Processo de Produção da Hitachi.....	369
--	-----

B.2.3 Processo de Produção da Toshiba.....	372
--	-----

B.2.4 Processo de Produção da NEC.....	374
--	-----

B.2.5 Processo de Produção da Fujitsu.....	376
--	-----

B.3 Gestão Operacional das Fábricas.....	378
--	-----

B.3.1 Gestão Operacional da SDC.....	379
--------------------------------------	-----

B.3.2 Gestão Operacional da Hitachi.....	379
--	-----

B.3.3 Gestão Operacional da Toshiba.....	381
--	-----

B.3.4 Gestão Operacional da NEC.....	385
--------------------------------------	-----

B.3.5 Gestão Operacional da Fujitsu.....	389
--	-----

B.4 As Ferramentas e Questões sobre o Armazenamento de Dados.....	392
---	-----

B.5 Produtos Gerados e Forma de Criação e Organização do Processo.....	396
--	-----

Anexo C – Requisitos Funcionais da Aplicação a ser Codificada e Testada pela Fábrica de Software da FATEC..... 400

Capítulo 1 – Introdução

Atualmente, o mercado de desenvolvimento de software brasileiro trava uma batalha constante na busca pela qualidade e produtividade. Esta informação pode ser comprovada ao analisar os vários programas de incentivo promovidos pelo Ministério de Ciência e Tecnologia (MCT), através da Secretaria de Política em Informática (SEPIN), na qual o governo estabeleceu na política industrial tecnológica que software é um dos quatro temas prioritários (software, semicondutores, indústria de base e fármacos). Entre tais programas, é possível destacar o SOFTEX (Sociedade para Promoção da Excelência do Software Brasileiro), cujos objetivos são: situar o Brasil entre os 5 (cinco) maiores produtores e exportadores de software do mundo e alcançar padrão internacional de qualidade e produtividade (maiores informações sobre o SOFTEX consulte www.softex.br).

Além destes programas, a SEPIN desenvolve, periodicamente, uma pesquisa para verificar os atributos da qualidade e produtividade do mercado brasileiro de desenvolvimento de software¹. A última delas foi publicada em 2006 (participaram da pesquisa 488 empresas e a margem de erro é de 2 pontos percentual para mais ou para menos) e, ao efetuar uma análise nos dados, é possível obter as seguintes informações: O Brasil conta com cerca de 16.000 estabelecimentos formais que prestam serviços ou desenvolvem produtos relacionados à área de informática (consultoria em hardware; desenvolvimento e edição de software pronto para uso próprio; desenvolvimento de software sob encomenda e outras consultorias em software; processamento de dados; atividade de banco de dados e distribuição on-line de conteúdo eletrônico; manutenção, reparação e instalação de máquinas de escritórios e de informática aglutinam as atividades das empresas).

Destes estabelecimentos, cerca de 2.86% encontram-se na região norte, 9.83% na região nordeste, 46.52% no sudeste, 35.25% no sul e 5.32% no centro-

¹ A pesquisa pode ser consultada na íntegra no endereço <http://www.mct.gov.br/index.php/content/view/3253.html> - acesso em novembro de 2006.

oeste. Se uma distribuição por estado for configurada é possível constatar que São Paulo (SP) aglutina 23.57% das empresas pesquisadas (veja uma comparação por estado na Tabela 1.1).

Estado	Número de Empresas	%
Rio Grande do Sul	81	16.6
Santa Catarina	62	12.5
Paraná	30	6.15
São Paulo	114	23.57
Rio de Janeiro	49	10.04
Minas Gerais	48	9.84
Espírito Santo	15	3.07
Bahia	14	2.87
Sergipe	2	0.41
Alagoas	0	0.00
Pernambuco	10	2.05
Paraíba	5	1.02
Rio Grande do Norte	1	0.20
Ceará	16	3.28
Piauí	0	0.00
Maranhão	0	0.00
Mato Grosso do Sul	2	0.41
Mato Grosso do Norte	3	0.61
Goiás	3	0.61
Distrito Federal	18	3.69
Tocantins	0	0.00
Acre	0	0.00
Amazonas	9	1.84
Rondônia	0	0.00
Pará	9	1.02
Roraima	0	0.00
Amapá	0	0.00

Tabela 1.1 – Número de empresas com atividades de informática divididas por estado (tabela delineada com base nas informações do site <http://www.mct.gov.br/index.php/content/view/3253.html>, acesso em novembro de 2006)

Além da distribuição por estado, a SEPIN delineou em sua pesquisa as empresas em micro (até 9 pessoas, a união de todas elas empregam cerca de 830 funcionários ou colaboradores), pequena (de 10 a 49 pessoas, a união de todas elas empregam cerca de 4.256 funcionários ou colaboradores), média (de 49 a 99 pessoas, a união de todas elas empregam cerca de 3.240 funcionários ou colaboradores) e grande (100 ou mais, empregando cerca de 97.300 funcionários ou colaboradores). Os resultados delineados para tal fato podem ser verificados por meio da Figura 1.1.

Figura 1.1 – Classificação, em %, das empresas em Micro, Pequena, Média e Grande (gráfico delineado com base nas informações do site <http://www.mct.gov.br/index.php/content/view/3253.html>, acesso em novembro de 2006)

Um outro dado interessante a ser analisado é produtividade das empresas em relação ao domínio de aplicação:

- 59% das empresas pesquisadas atuam na comercialização ou desenvolvimento de pacotes (software de prateleira);
- 81.1% desenvolvem software sob encomenda;
- 18.2% desenvolvem software embarcado;
- 36.3% desenvolvem software para uso próprio.

Ressalta-se que os números apresentados nos itens acima se sobrepõem, por exemplo: empresas podem desenvolver software sob encomenda e pacote dentro de sua carteira de produtos.

Além do domínio de aplicação, a SEPIN consolidou os resultados da pesquisa em segmentos produtivos relacionados à atividade de TI. Os dados sobre este aspecto são apresentados na Tabela 1.2.

Áreas relacionadas a TI em que a organização atua, além das atividades de software.	
Segmentos produtivos	%
Automação Bancária	9.2
Automação Comercial e de Serviços	43.0
Automação Industrial	19.7
Computadores e periféricos	9.8
Instrumentação médico-hospitalar	2.0
Telecomunicações – computação	3.9
Telecomunicações – transmissão	8.0
Telecomunicações – terminais	4.3
Acesso à internet	16.6
Assistência técnicas em hardware	11.3
Comércio eletrônico	21.5
Comunicação de dados	18.6
Consultoria em projetos em TI	63.1
Pesquisa e desenvolvimento em TI	43.0
Processamento de dados	21.7
Serviços tecnológicos	36.5
Treinamento em TI	34.0
Outras	0.6
Não atua em outras áreas relacionadas a TI	8.6

Tabela 1.2 - Áreas relacionadas a TI em que a organização atua além da atividade de software. Existência de uma sobreposição nos dados. (tabela delineada com base nas informações do site <http://www.mct.gov.br/index.php/content/view/3253.html>, acesso em novembro de 2006)

A origem do capital das organizações é um outro dado interessante, e foi capturado pela pesquisa em questão. Cerca de 96% das empresas pesquisadas tiveram investimentos do capital privado, destas, 93% estruturam-se com o capital nacional, 1.2% tiveram como alicerce embrionário o capital estrangeiro e, 1.9% se caracterizaram com o capital misto (nacional e estrangeiro). Apenas 3.9% das empresas, se estruturaram com o capital público, deste índice, 1.0% é caracterizado como investimento do governo federal, 2% dos governos estaduais e 0.9 dos governos municipais.

Além dos dados que caracterizam as empresas, a página da SEPIN (www.mct.gov.br/sepin) apresenta a quantidade de empresas com certificação em qualidade do processo. Os resultados estão consolidados da seguinte forma:

1. **A norma ISO 9001²** em âmbito mundial, continental, nacional e regional. Este trabalho apresenta os números de certificações, emitidos pela ISO, para qualquer tipo de empresa, inclusive para o setor de produção de software – veja Tabela 1.3, Tabela 1.4 e Tabela 1.5. Uma visão sobre a quantidade de empresas brasileiras certificadas na norma ISO 9001, em relação à área de informática e conexas, também, pode ser verificada na Tabela 1.6.

CERTIFICAÇÕES ISO 9001 POR CONTINENTE		
Continente	Total de empresas certificadas	%
América Central	371	0.07
África	4.465	0.80
América do Sul	13.306	2.36
América do Norte	53.806	9.58
Ásia	167.450	29.83
Europa	292.998	52.16
Oceania	29.204	5.20
Total	561.600	100
Países com o maior número de empresas certificadas		
China	75.755	13.49
Reino Unido	60.960	10.85
Estados Unidos	38.927	6.93
Alemanha	35.802	6.38
Japão	33.964	6.05
Total	245.408	86.27

Tabela 1.3 – Certificados ISO 9001 emitidos no mundo (classificação por continente) – situação até 2004 (Acessado a partir de www.mct.gov.br/sepim – Fonte original, segundo a SEPIM, site da ISO <http://www.iso.ch/iso/en/prods-services/otherpubs/pdf/survey13thcycle.pdf>).

² ISO: *International Organization for Standardization*. A ISO é uma organização não governamental e caracteriza-se como uma rede de institutos de padronização. Atualmente, a organização é composta por 157 países, sendo que seu comando central fica localizado em Genebra (Suíça). O principal objetivo da organização é fornecer padrões e modelos para a padronização de diversos setores produtivos da sociedade, destaque para os seguintes setores: governo; consumidores de bens e serviços e; empresas. A ISO 9001 é caracterizada como uma referência internacional para os requisitos da qualidade de uma empresa. A norma em questão direciona o processo de gestão de uma empresa e seus principais objetivos são: 1 – Tratar requisitos da qualidade para o cliente aumentando, assim, sua satisfação. 2 – Prover um mecanismo ininterrupto de melhoria de desempenho em relação aos objetivos da organização (fonte: <http://www.iso.org/iso/en/aboutiso/introduction/index.html> - consultada em dezembro de 2006).

CERTIFICAÇÕES ISO 9001 POR PAÍS DA AMÉRICA DO SUL			
País da América do Sul	Total de empresas certificadas	%C [^]	%M [^]
Argentina	2.260	16.98	0.402
Bolívia	31	0.23	0.005
Brasil	7.900*	59.37	1.407
Chile	327	2.46	0.058
Colômbia	1.838	13.81	0.327
Equador	34	0.26	0.006
Guiana	7	0.05	0.001
Paraguai	65	0.49	0.012
Peru	270	2.03	0.048
Suriname	1	0.01	0.0002
Uruguai	231	1.74	0.041
Venezuela	342	2.57	0.061
Total	13.306	100	2.3682
Legenda: %C = percentual em relação ao continente (base para calculo 13.306) %M = percentual em relação ao mundo (base para calculo 561.600). As variáveis %C e %M não estão presentes na tabela original. *Indica o número de empresas certificadas, porém no Brasil foram emitidos 8175 certificados ISO 9000. ^Calculo arredondado em duas e três casas decimais			

Tabela 1.4 - Certificados ISO 9001 emitidos por países da América do Sul – situação até 2004 (Acessado e adaptado a partir de www.mct.gov.br/sepin – Fonte original, segundo a SEPIN, site da ISO <http://www.iso.ch/iso/en/prods-services/otherpubs/pdf/survey13thcycle.pdf>)

Estados da Federação	TOTAL
Acre	3
Alagoas	34
Amazonas	161
Amapá	1
Bahia	272
Ceará	88
Distrito Federal	143
Espírito Santo	188
Goiás	157
Maranhão	34
Minas Gerais	642
Mato Grosso do Sul	25
Mato Grosso	30
Pará	43
Paraíba	23
Pernambuco	168
Piauí	5
Paraná	580
Rio de Janeiro	542
Rio Grande do Norte	37
Rondônia	6
Roraima	1
Rio Grande do Sul	605
Santa Catarina	389
Sergipe	18
São Paulo	3972
Tocantins	8
Total	8175

Tabela 1.5 – Números de certificações ISO emitidos por estado da federação (tabela delineada com base nas informações do site <http://www.mct.gov.br/index.php/content/view/3253.html>, acesso em novembro de 2006)

Certificações ISO 9001 para as empresas com atividade na área de informática e conexas	
Consultoria relativa a hardware	16
Consultoria e fornecimento de software (1,20%)	98
Processamento de dados	10
Atividade de banco de dados	3
Manutenção e conserto máquinas de escritório, de contabilidade e de informática.	28
Outras atividades conexas a informática	76
Total (2,83% das empresas com atividade na área de informática são certificadas ISO 9001)	231

Tabela 1.6 - Certificações ISO 9001 para as empresas com atividade na área de informática e conexas – situação até junho de 2006 (tabela delineada com base nas informações do site <http://www.mct.gov.br/index.php/content/view/3253.html>, acesso em novembro de 2006)

2. Os modelos CMM³ (*Capability Maturity Model*) e CMMI⁴ (*Capability Maturity Model Integration*) em âmbito continental e nacional. Porém, o autor deste trabalho apresenta a consolidação dos números, nos itens destacados abaixo deste parágrafo, em âmbito mundial (os dados apresentados neste item foram coletados do Site do SEI (*Software Engineering Institute*: <http://www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/2006sepCMMI.pdf>). Segundo o SEI de abril de 2002 a julho de 2006, foram efetuadas 1581 avaliações, em relação ao modelo CMMI, sendo que 63.8% das empresas avaliadas não são americanas. Em relação aos níveis de maturidade das organizações é perceptível que:

- 1.9% encontram-se no nível inicial;
- 33.3% no nível gerenciado;
- 33.8% no nível definido;

³ O CMM - *Capability Maturity Model* é um modelo para avaliação da maturidade dos processos de software de uma organização e para identificação das práticas-chave que são requeridas para aumentar a maturidade destes processos. Ele prevê cinco níveis de maturidade: inicial, repetível, definido, gerenciado e otimizado. O modelo foi proposto por Watts S. Humphrey, a partir das propostas de Philip B. Crosby, e vem sendo aperfeiçoado pelo *Software Engineering Institute* – SEI da Carnegie Mellon University. Para maiores informações sobre o modelo consulte www.sei.cmu.edu/cmm/cmm.html. Salienta-se que o CMM foi substituído pelo CMMI.

⁴ O CMMI - *Capability Maturity Model Integration* foi criado pelo SEI como uma integração e evolução dos modelos: SW-CMM - *Capability Maturity Model for Software*; SECM - EIA 731 - *System Engineering Capability Model*, e IPD-CMM - *Integrated Product Development CMM*. O CMMI é um modelo alinhado com a Norma ISO/IEC 15504 e é apresentado em duas representações: uma por estágio (como o CMM), e outra contínua (semelhante à ISO/IEC 15504).

- 4.4% no nível gerenciando quantitativamente;
- 18.2 no nível otimizado.

Em relação ao tamanho das organizações, o SEI mostra que 43% das avaliações foram efetuadas em empresas que possuem de 1 a 100 funcionários; 20% em empresas que possuem de 101 a 200 e; 37% em empresas que possuem mais de 201 funcionários (vide Tabela 1.7)

Tamanho das Organizações	% de avaliações
1 a 25	9,8
26 a 50	13,4%
51 a 71	11,2%
76 a 100	8,7%
Total	43,0%
101 a 200	20,0%
Total	20,0%
201 a 300	10,9%
301 a 500	9,5%
501 a 1000	8,2%
1001 a 2000	5,6%
2000 ou mais	2,8%
Total	37,0%

Tabela 1.7 – Certificação CMMI em âmbito mundial, em relação ao tamanho das empresas.

Uma outra informação interessante relatada pelo instituto é o número de organizações avaliadas por país. Ao analisar tais dados é possível verificar que o Brasil encontra-se na oitava posição (vide Tabela 1.8).

Uma outra pesquisa que caracteriza a afirmação sobre a busca constante da qualidade e produtividade do setor de software brasileiro, apresentada no início deste capítulo, foi desenvolvida pela Associação Brasileira das Empresas de Software e Serviços para Exportação (*Brazilian Association of Software & Service Export Companies* – BRASSCOM) com assessoria da A.T. Kearney (consultoria especializada no desenvolvimento de pesquisas de mercado para a área de Tecnologia da Informação). Os dados de tal pesquisa foram publicados em dezembro de 2005 e podem ser obtidos no site da BRASSCOM

(<http://www.brasscom.com.br/>). Salienta-se que o objetivo principal da pesquisa é mapear as condições para que o Brasil possa exportar serviços de TI.

País	Número de Organizações certificadas
Estados Unidos	598
Índia	177
China	158
Japão	155
França	65
Coréia do Sul	56
Reino Unido	42
Brasil (8º)	39
Taiwan	31
Alemanha	28
Espanha	25
Austrália	23
Canadá	18
Argentina	15
Malásia	15
Filipinas	14
Egito	10
África do Sul	Menos que 10
Austria	
Bahrain	
Belarus	
Bélgica	
Chile	
Colômbia	
Dinamarca	
Eslováquia	
Finlândia	
Holanda	
Hong Kong	
Indonésia	
Irlanda	
Israel	
Itália	
Latvia	
Marrocos	
Mauritius	
México	
Nova Zelândia	
Paquistão	
Portugal	
República Dominicana	
República Tcheca	
Rússia	
Singapura	
Suécia	
Suíça	
Turquia	
Ucrânia	
Vietnã	

Tabela 1.8 – Certificações CMMI em nível mundial. A organização da tabela foi adaptada do Site do SEI: <http://www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/2006sepCMMI.pdf>

A pesquisa desenvolvida pela BRASSCOM, mostra que o mercado, de *outsourcing*⁵ e de Tecnologia da Informação (TI), é desenvolvido na América do Norte e na Europa, onde se concentra a demanda por serviços de maior valor agregado (aplicativos e processo de negócio). Juntos, estes continentes representam quase 80% da demanda por *outsourcing* e de serviços de TI (vide Figura 1.2)

Figura 1.2 - Distribuição geográfica do mercado global de *outsourcing* de serviços de TI (US\$ Bn/ano [Bilhão de dólares por ano] – 2004). Retirado do relatório sumarizado da pesquisa publicada pela BRASSCOM, página 5.

Um outro dado interessante apontado pela pesquisa é sobre o crescimento do mercado de *outsourcing* e serviços de TI de 2002 a 2006. Segundo os dados publicados pela BRASSCOM, em 2004, o serviço de TI totalizou US\$ 607 bilhões e a perspectiva de crescimento é de 6% ao ano até 2008.

Ao analisar a Figura 1.3, é possível verificar que o segmento de aplicativo representa o maior investimento na área de TI. Em 2004 as cifras para tal segmento

⁵ Segundo THO(2005), a ideia de *outsourcing* pode ser caracterizada pelo seguinte exemplo: Uma determinada empresa, localizada no país A, contrata outra empresa, localizada no país B para a produção de código fonte.

chegaram a US\$ 239 bilhões⁶. Porém, os processos de negócio deverão apresentar o maior crescimento até 2008, com 8,2% de taxa anual.

Figura 1.3 - Evolução do mercado global de *outsourcing* por segmento de escala de serviços de TI (US\$ Bi/ano – 2004). Retirado do relatório sumarizado da pesquisa publicada pela BRASSCOM, página 6. Neste trabalho não foi verificado se as previsões de 2006 foram consolidadas.

Vários países se qualificam para capturar uma parcela expressiva do mercado analisado na pesquisa (vide Figura 1.4). Os competidores do Brasil neste cenário podem ser divididos em duas categorias.

- Países tradicionais exportadores de serviços de TI (ex. Canadá, Índia e Irlanda);
- Países exportadores emergentes (ex. China, Malásia, Filipinas, países do Leste Europeu e México).

⁶ Os aplicativos apresentam grande volume, devido a grande demanda por serviços de desenvolvimento e de integração de novos sistemas para processo de negócio e modernização e adequação de sistemas legados.

Figura 1.4 - Tamanho dos principais mercados de software e serviços de TI – 2003 (US\$ Milhões). Retirado do relatório sumarizado da pesquisa publicada pela BRASSCOM, página 6.

Ao analisar a Figura 1.4 é possível verificar que o mercado de TI brasileiro encontrava-se, em 2004, na quinta colocação sob a ótica da capitalização monetária, perdendo somente para a Irlanda, China, Índia e o Canadá. Um fato chama a atenção na figura: grande parte dos valores agregados no Brasil era para atender o mercado interno, o que leva a crer que o setor de produção de software contribuiu para o baixo índice de exportação de serviços de TI. Crença esta que pode ser comprovada ao analisar uma pesquisa desenvolvida pela consultoria IDC, que mostra que 71% dos programas de computação usados no Brasil, em 2005, foram importados e, apenas 1.2% de produtos e serviços relacionados a software foram exportados no mesmo ano (fonte <http://agenciact.mct.gov.br/index.php/content/view/40141.html>, publicação de 29/06/2006). Os dados publicados pela IDC, também, mostram que o país exportou cerca de US\$ 178 milhões. Neste mesmo ano, o montante monetário, no setor de software, arrecadado no Brasil chegou a US\$ 7.41 bilhões.

Para encerrar a apresentação dos dados, que validam a afirmação apresentada nas primeiras linhas deste capítulo, serão destacadas as palavras do

presidente da Associação Brasileira das Empresas de Software (ABES), Jorge Sukarie Neto, em relação ao setor de software no Brasil:

“A verdade é que continuamos com os velhos problemas ligados à qualificação de mão-de-obra, tributação, certificação das empresas, legislação trabalhista e financiamento”.

(fonte <http://agenciact.mct.gov.br/index.php/content/view/40141.html>, publicação de 29/06/2006).

Com base no contexto apresentado é possível afirmar que existe um paradoxo dentro do setor de software Brasileiro. Dentro do contexto mundial somos o quinto na capitalização monetária em TI (contexto dos países emergentes). Em contrapartida exportamos, apenas, 1.2% dos produtos relacionados a software, setor este que agregará cerca 39% do mercado de TI, em 2008 (vide Figura 1.3), totalizando 2.4% em valores monetários comercializados, nacionalmente. Em relação a certificações de processo (CMMI e ISO 9001) é visto que:

- O Brasil ocupa o oitavo lugar em números de certificações CMMI no mundo. Porém existe um grande caminho a percorrer para alcançarmos Índia e China;
- Na área de consultoria e fornecimento de software o Brasil possui 1.20% das empresas com certificação ISO 9001;
- Somos líderes no número de certificações em processos na América do Sul, porém o continente agrega 3% do capital em *outsourcing* e serviços de TI (vide Figura 1.2).

Enfim, analisando o contexto paradoxal, pode-se considerar que o país pode vir a atingir (a médio e longo prazo) os objetivos traçados pelo SOFTEX: situar o Brasil entre os 5 (cinco) maiores produtores e exportadores de software do mundo e alcançar padrão internacional de qualidade e produtividade. Para isto, é necessário um esforço em conjunto (a médio e longo prazo) do Governo (melhorando o sistema tributário, incentivando a exportação de produtos relacionados a TI), das Empresas (conscientizando sobre a necessidade de certificação em modelos de processos,

como o CMMI) e das Universidades (dando ênfase em aspectos da qualidade e produtividade nas disciplinas do núcleo de engenharia de software, despertando, assim, o caráter empreendedorista nos alunos) dentro do tema produtividade, qualidade do processo e do produto software, temas estes quando unidos podem se configurar como uma **FÁBRICA DE SOFTWARE** (SPÍNOLA et. al. (2004) e FABRI et. al. (2006)).

O quadro apresentado evidencia a necessidade do aprofundamento nas questões referentes à confecção de software com qualidade, a fim de permitir a inserção do país no contexto mundial de produção de bens e serviços ligados à área de tecnologia e sistemas da informação (TI e SI). É somente com estruturas que possibilitem a organização ou a criação de processos de software que isto pode vir a acontecer.

1.1 Objetivos e Caracterização do Trabalho

As últimas palavras da seção anterior afirmam que é necessária a união de três pilares da sociedade para que o Brasil possa aumentar a produção dentro do setor de software: Governo, Empresa e Universidade. Este trabalho foca dois deles: Universidade e Empresa.

- No pilar universitário o trabalho tem como objetivo verificar:
 1. Propor um modelo que possibilite a criação ou organização de um processo de software com características fabris. Este objetivo é caracterizado como principal e os demais podem ser caracterizados como objetivos secundários.

2. Se o modelo fabril para produção de software, proposto pelo ELABSOFT⁷ é aderente às fábricas de software atuantes no mercado brasileiro. Para isto foram realizados 6 estudos de casos em fábricas, 5 delas certificadas no modelo CMMI e uma na norma ISO 9001.
 3. Se o modelo, proposto pelo ELABSOFT, adere aos casos clássicos da literatura (Toshiba, Hitachi, NEC, Fujitsu e a americana SDC).
 4. Verificar se os processos fabris brasileiros, nipônicos e americano estudados aderem ao modelo.
 5. Verificar se o processo de produção de software das fábricas brasileiras (pesquisadas) é definido por meio de uma estrutura formal (métodos, procedimentos que levem a estruturação do processo de produção).
 6. Verificar se existe um processo genérico de produção de software em um ambiente fabril.
 7. Verificar se as empresas que se denominam como fábrica de software podem ser descritas como fábrica de programas ou fábrica de projetos de software (a definição de tais termos pode ser encontrada no capítulo 2).
- No pilar empresarial, o modelo foi aplicado por uma organização do setor produtivo de software, caracterizada como fábrica, na definição de um processo de produção de uma de suas parceiras. Tal aplicação tem como objetivo verificar a aplicabilidade do modelo proposto neste trabalho.

Definidos os objetivos, principais e secundários, e efetuada a caracterização mercadológica do trabalho, a próxima seção apresenta, de forma compacta, a

⁷ O ELABSOFT é uma das linhas de pesquisa do Grupo de pesquisadores em Tecnologia da Informação (GTI) do Departamento de Engenharia de Produção da Escola Politécnica da Universidade de São Paulo. Um dos objetivos deste grupo é desenvolver uma “fábrica modelo” para a produção de software. Tal pesquisa objetiva conceituar, métodos, técnicas e ferramentas relativas ao contexto fabril para software, na forma mais ampla possível. Uma descrição detalhada de tal grupo pode ser vista no capítulo 3.

estrutura metodológica para o desenvolvimento do trabalho. Uma apresentação completa sobre os aspectos metodológicos relacionados a este trabalho pode ser verificado no capítulo 6.

1.2 Estrutura Metodológica para o Desenvolvimento do Trabalho

A estrutura metodológica para o desenvolvimento deste trabalho está baseada nos pressupostos de YIN (2005) e nos relatos publicados por FABRI (2005a). Ambos os autores apresentam um seqüência de passos, enumerados a seguir, para o desenvolvimento de um trabalho de científico.

1. Definir uma questão de estudo: a questão de estudo direciona a pesquisa em si, estabelecendo assim, o foco na qual o pesquisador irá imergir. A seleção de uma questão demanda conhecimentos sobre a área da ciência na qual se deseja navegar. A percepção das necessidades da humanidade, também, pode ajudar na definição da questão de estudo. Sendo assim, é pertinente afirmar que a questão a ser respondida neste trabalho é: É possível construir um modelo que possibilite a criação ou organização de um processo de software com características fabris? Esta questão está, intimamente, ligada com o objetivo principal do trabalho, delineado na seção anterior.
2. Estabelecer as proposições: uma proposição direciona algum fato que deva ser examinado dentro da pesquisa. Ressalta-se que uma proposição pode ser confirmada ou não, após a realização da pesquisa. Este trabalho enumera três proposições:
 - a. O processo de produção de uma fábrica de software é definido por meio de uma estrutura formal (métodos, procedimentos que levem a estruturação do processo de produção). Estes métodos são aplicados por tais empresas?

- b. Existe um processo genérico de produção de software que pode ser aplicado ao contexto fabril.
- c. O ciclo de produção da uma fábrica de software é classificado como curto, médio e/ou longo.

Salienta-se que as três proposições, que direcionam este trabalho, também, aderem aos objetivos traçados na seção anterior.

3. Definir o método de pesquisa: a definição do método de pesquisa está ligada à questão que se deseja responder. YIN (2005) ressalta, ainda, que a escolha do método está relacionada a duas condições básicas: controle que o pesquisador possui sobre os eventos comportamentais efetivos e o foco em fenômenos históricos, em oposição a fenômenos contemporâneos. Este trabalho utiliza a técnica de estudo de caso como método de pesquisa. Tal estudo é dividido em duas etapas: A primeira relata a estrutura do processo e produção de software das fábricas brasileiras. Já a segunda relata como o modelo proposto foi aplicado na definição de um processo de produção de software. Detalhes sobre o método de pesquisa utilizado será apresentado no capítulo 5.
4. Delinear as unidades de análises: Quais serão as unidades (experimentos, casos, simulações) que devem ser mapeadas e analisadas? Doze casos, seis fábricas brasileira e cinco fábricas estrangeiras compõem as unidades de análise. A empresa que se propôs a aplicar o modelo apresentado neste trabalho constitui o décimo segundo caso. É importante salientar que as informações dos casos estrangeiros foram colhidas de CUSUMANO (1991).
5. Relacionar os dados gerados nas unidades de análise e interpretar as constatações: De posse dos dados gerados nas unidades de análise o pesquisador deve inferir e desenvolver constatações, contribuindo assim, com a evolução científica.

Por fim, é importante ressaltar que um relato detalhado sobre os procedimentos metodológicos utilizados para o desenvolvimento deste trabalho é efetuado no capítulo 6.

1.3 Composição Estrutural do Trabalho

Além deste capítulo introdutório, o trabalho possui mais outros dez.

O capítulo 2 apresenta uma revisão bibliográfica sobre fábrica de software. Neste capítulo, também, é apresentado os casos estrangeiros, delineados após uma análise da literatura.

Uma evolução do modelo fabril para produção de software proposto pelo ELABSOFT é documentada no capítulo 3.

Uma revisão bibliográfica sobre processo de software e os mecanismos e modelos que proporcionam a organização ou criação de processos estão descritos no capítulo 4.

O capítulo 5 apresenta a proposta preliminar do Modelo para a Criação e a Organização de Processos de Produção em um Contexto de Fábrica de Software. É valido ressaltar que tal proposta efetuada está alicerçada nas considerações apresentadas nos capítulos 2 e 4.

O capítulo 6 apresenta a questão que norteia o desenvolvimento do trabalho, as proposições, o método de pesquisa, o protocolo para o desenvolvimento da pesquisa e as unidades de análises. As justificativas para a escolha do método estudo de caso, também, são delineadas.

As informações colhidas na pesquisa de campo são apresentadas, de forma condensada, no capítulo 7. Já o anexo A apresenta o seu detalhamento.

Uma análise comparativa entre as empresas estrangeiras e brasileiras é apresentada no capítulo 8. Sienta-se que os aspectos utilizados para efetuar tal comparação são delineados pelos capítulos 2 e 3.

O capítulo 9 mostra a aderência dos casos, brasileiros e estrangeiros, ao modelo proposto no capítulo 5. Ao realizar a aderência, alguns ajustes no modelo foram efetuados.

Um experimento prático do modelo é apresentado no capítulo 10. A aplicabilidade do modelo na definição e organização de um processo real é verificada.

Por fim, o capítulo 11 apresenta a conclusão do trabalho, as proposições são confirmadas ou não, e sugestões para se configurar novas linhas de pesquisa, a partir deste trabalho, são delineadas.

Capítulo 2 – Fábrica de Software

Atualmente, vários pesquisadores trabalham com o tema fábrica de software. No Brasil, uma das primeiras obras literárias para tal segmento foi desenvolvida por FERNANDES e TEIXEIRA (2004). Ambos autores conclamam que, “uma fábrica de software pode ter vários domínios de atuação, desde um projeto de software completo, até um projeto físico ou a codificação de um programa de computador (página 115)”. Os autores enumeram uma série de atributos inerentes a uma estrutura fabril para software (vide Quadro 2.1).

Quadro 2.1 - Atributos para uma fábrica de software (Adaptado de FERNANDES e TEIXEIRA 2004)

Processo definido para o desenvolvimento de software.
 Alto poder de atendimento.
 Ordens de serviços padronizadas.
 Estimativas de custo e prazo.
 Controle dos perfis de recursos humanos.
 Processo de planejamento e controle da produção.
 Controle de componentes ou itens de software.
 Produção de software deve estar fortemente baseada em métodos e técnicas padronizadas.
 Treinamento de recursos humanos.
 Garantia da qualidade do produto.
 Melhoria contínua do processo.
 Ambiente de software e hardware deve estar alinhado com as necessidades do usuário.

Já no contexto internacional, este trabalho destaca as pesquisas efetuadas por FERNSTROM et. al. (1992), CANTONE (1992), BASILI et. al. (1992), RUHE (1999), LI et. al. (2001) e HUMPHREY (2001).

FERNSTROM et. al. (1992), definem que uma organização fabril para o desenvolvimento de software deve estar calcada na questão “única do software”, isto é, todo software é único, porém partes individuais são repetidas em vários projetos. O conceito fabril deve alicerçar o desenvolvimento, armazenamento e montagem das partes repetidas em um produto único.

Já CANTONE (1992) salienta que uma fábrica de software deve:

- ser flexível, capaz de produzir vários tipos de produtos dentro de um segmento de mercado;
- implementar os conceitos de engenharia de software (metodologia, ferramental, configuração do ambiente) e, também;
- ser capaz de estudar, projetar, implementar, evoluir e melhorar os sistemas.

O autor apresenta, em seu trabalho, um modelo evolucionário, que deve ser observado na composição de estruturas fabris para software, cujo objetivo é garantir o valor produtivo e qualitativo implícito nelas. A visualização do modelo, proposto por CANTONE (1992), pode ser observada na Figura 2.1.

Figura 2.1 – Modelo evolucionário (orientado a qualidade) adaptado de CANTONE (1992)

Legenda: PPM = Modelo de produto e processo (*product and process models*). SF MODELS = Modelos de fábrica de software (*software factory model*). SF = Fábrica de software (*software factory*) instanciada. PRODUCT = Produtos. QI = Melhoria da Qualidade (*quality improvement*). SW QL = qualidade de software (*software quality*). Nota: o formato da figura segue a proposta original da figura.

Ao analisar a Figura 2.1 é possível perceber que uma fábrica de software é instanciada a partir do compartimento SF MODEL (modelos de fábricas de software). Segundo o autor, tais modelos são configurados a partir de experiências bem sucedidas nas empresas produtoras de software, relatadas pela literatura. A fábrica de software deve ser orientada ao produto e, conseqüentemente, ao processo, ela deve focar domínios do conhecimento para se consolidar, isto é, uma fábrica deve desenvolver produtos em alguns segmentos, por exemplo: desenvolvimento de

aplicações móveis¹ (L'ERÁRIO et. al (2004)). O compartimento de modelos da qualidade para software (SW QL) direciona a concepção dos modelos de processos e de produtos. Por fim, o compartimento de melhoria da qualidade (QI) envolve os modelos da qualidade com o objetivo de melhorar o processo de software, implementado.

Um outro modelo orientado aos conceitos de qualidade foi proposto por LI et. al. (2001) e apresentado na Figura 2.2. Além da proposta do modelo, os autores definem que uma fábrica de software deve possuir um conjunto de ferramentas padronizadas para construção de software, bases históricas a serem utilizadas no gerenciamento de projetos e, principalmente, um alto grau de reuso de código no processo de produção de um determinado software.

No modelo, proposto por LI et. al. (2001), é possível observar a presença de 5 entidades bem definidas:

- Entidade 1 – Técnicas: Responsável por contextualizar as tecnologias emergentes para configurar o processo de desenvolvimento de software, as ferramentas e os componentes. Técnicas de reuso também fazem parte desta entidade;
- Entidade 2 – Processo: Definição das atividades e tarefas do processo de produção de software. Nesta entidade também é possível verificar a atividade de gerenciamento e controle que evolui, paralelamente, a todas as atividades do processo de produção de software;
- Entidade 3 – Envolvidos: Pessoas que atuam, diretamente, no desenvolvimento do software;

¹ A idéia de segmentação da produção de uma fábrica de software também é compartilhada por FABRI et. al. (2005).

Figura 2.2 – Modelo orientado à qualidade (norma ISO 9000 e modelo CMM) (adaptado de LI et. al. (2001))

- Entidade 4 – Especificação de gerenciamento: Responsável por definir a estrutura organizacional da fábrica de software, o processo fabril e o gerenciamento da qualidade. Nesta entidade são definidas as especificações gerenciais praticadas na entidade 2;
- Entidade 5 – Ativos de processo, ferramentas e componentes de código: Entende-se como ativo de processo formulários, padrões, algoritmos utilizados como artefato no processo.

Na Figura 2.2 é possível verificar, também, que a entidade “técnicas” provê o suporte técnico/conceitual para a definição do processo. O processo é norteador pelo

modelo CMM (relação direta com o modelo proposto por CANTONE (1992)) e os requisitos da qualidade para a organização fabril são definidos pela norma ISO 9001. Segundo os autores, a ISO 9001 é uma norma utilizada no contexto industrial e seu foco está no sistema de qualidade da organização, fato este que motiva a inserção da norma no modelo. Já o CMM é destinado para a indústria de software, deste modo as áreas chaves provêm maiores detalhes para a configuração do processo, motivo pelo qual os autores inseriram tal documento no modelo. Conclui-se, então, que os autores usam ISO 9001 como base de gerenciamento organizacional e CMM como base para configuração e gerenciamento de projetos e processos de software.

A entidade “especificação de gerenciamento” define, através das sub-entidades “gerenciamento da qualidade” e “organização do modelo de processo”, os atores (envolvidos no processo de desenvolvimento de software) e as suas responsabilidades. A entidade “envolvidos” é norteadada pelos conceitos advindos do *Personal Process Software* (PSP) (HUMPHREY (2005a)) e do *Team Process Software* (TSP) (HUMPHREY (2005)). Salienta-se, também, que a sub-entidade “organização do modelo de processo” define características do processo de desenvolvimento de software.

Por fim, os ativos do processo, as ferramentas e os componentes de código dão suporte ao processo de desenvolvimento de software (vide entidade 5 da Figura 2.2).

Um fato interessante deve ser explicitado: ao analisar o modelo proposto por LI et. al. (2001) é perceptível sua preocupação com gestão do negócio para uma fábrica de software (vide entidade 4, na qual esta foca conceitos da estrutura organizacional). Este fato levou TRINDADE (2006) a associar o modelo proposto por LI et. al. (2001), com o modelo composto de categorização por níveis, configurado na Figura 2.3. Tal associação é verificada na Figura 2.4.

Na Figura 2.3 é possível notar a presença de duas visões em um contexto organizacional: a gerencial e a processual. Na visão gerencial nota-se os níveis: estratégico, tático e operacional, cada qual com suas responsabilidades. Na visão processual é possível perceber o compartimento caracterizando o processo transformador, o qual executa tarefas de produção industrial, comercial ou intelectual.

Figura 2.3 – A associação proposta entre as visões gerencial e processual. Figura apresentada por TRINDADE (2006) página 39.

Na Figura 2.4 é verificada a presença de duas composições: estática (baixo índice de mudança dentro de um contexto organizacional) e dinâmica (alto índice de mudança dentro de um contexto organizacional). No âmbito estático é possível verificar as entidades: quatro e cinco (LI et. al. (2001)). A quarta entidade trata as questões de gerenciamento organizacional da fábrica de software, incluindo qualidade e organização estrutural do processo². Nota-se que a entidade quatro abrange os níveis estratégico e tático. Já a entidade cinco está posicionada no nível operacional cuja responsabilidade é prover o ferramental para as execuções das

² Organização estrutural do processo mapeia as características globais de um processo de desenvolvimento de software, classificando-o segundo a sua categoria ou modelo (cascata, incremental, orientado a reuso, são exemplos de categorias de processo). (SOMMERVILLE (2003 e 2006))

melhoria profissional³. A fábrica constitui um processo intensivo em mão de obra e, portanto, sua gestão deve focar defeitos na produção, não como uma forma de punição aos profissionais, mas como uma oportunidade de melhoria do processo;

- Princípio 2 – Metodologia de processo: Os processos devem ser, formalmente, definidos e ter elementos que provejam suporte à mensuração, para que dados estatísticos possam ser colhidos e, analisados com o objetivo de identificar problemas e determinar as causas dos mesmos;
- Princípio 3 – Suporte ao processo. Prover infra-estrutura necessária para que grupos de profissionais sejam especializados e treinados, tanto no âmbito produtivo como no administrativo, objetivando, assim, o bom uso do ferramental e dos ativos de processo presentes em um ambiente de produção de software com características fabris;
- Princípio 4 – Controle de processo e melhoria contínua. Boas práticas de gerenciamento são bem vindas e necessárias ao controle de mudanças, considerando que a avaliação dos períodos produtivos é conduzida para o monitoramento de efetividades e mapeamento de necessidades de melhoria, fato este que leva ao estabelecimento e implementação de ações corretivas, quando necessárias, e, conseqüentemente, da qualidade do processo.

Além de HUMPHREY (2001) e TRINDADE (2006), RUHE (1999) apresenta em seus estudos, a aproximação da produção de software com as práticas administrativas, salientando que o produto software possui características incomuns em relação ao gerenciamento, pois exige uma concepção organizacional do trabalho capaz de aliar a dimensionalidade múltipla de um trabalho intelectual com a diretriz linear concebida em uma organização fabril. Em sua proposta, RUHE (1999) se baseia em um modelo de mensuração orientado a metas, denominado de paradigma GQM (*Goal-Question-Metrics*).

³ A administração de pessoal está relacionada à valorização dos fatores humanos, presente na maioria dos processos de produção.

BASILI et. al. (1992) foge um pouco das características organizacionais focadas pelos autores, anteriormente, citados e trata o conceito de fábrica de software sob o ponto de vista da organização do processo de produção. Para tais autores, o processo fabril para a produção de software é segmentado em:

1. Processo de produção de componentes.
2. Processo de produção de software baseada em projetos.

A fábrica de componentes tem como objetivo produzir componentes, hermeticamente, fechados que devem ser utilizados nas unidades de software⁴ montadas pela fábrica de software. A visão embrionária da segmentação de processo proposta pelos autores do modelo pode ser verificada na Figura 2.5.

Figura 2.5 - Visão embrionária de uma fábrica de software com processo segmentado. Adaptada de BASILI et. al. (1992)

De posse desta visão BASILI et. al. (1992) propuseram, na Universidade de Maryland, o conceito de “*fábrica experimental*”. Tal conceito pode ser caracterizado como uma evolução da proposta materializada na Figura 2.5 (vide Figura 2.6). No processo de evolução é possível verificar as solicitações de produtos (componentes para construção de software), de dados (estatísticas para controle de custo, prazo e qualidade) e de planos (modelos, métodos para análise e projeto de software). Após

⁴ Uma unidade de software pode ser caracterizada como um programa executável, uma rotina ou função ou, ainda, um componente utilizado na composição do software como um todo.

a solicitação, a organização baseada em projetos recebe, da fábrica de componentes, os modelos e os componentes para construção do software.

Figura 2.6 - Fábrica experimental (evolução da visão embrionária) - Adaptação de BASILI et. al. (1992)

Ressalta-se, também, a presença da base de componentes na Figura 2.6, cujo objetivo é armazenar os componentes desenvolvidos pela fábrica, e as experiências adquiridas na execução de projetos, neste último caso a base de componentes se comporta como uma base de experiência de projetos.

É importante salientar que o modelo fabril para produção de software proposto por BASILI et. al. (1992) pode ser customizado, as atividades da organização baseada em projeto (ou fábrica de software) e da fábrica de componentes podem ser estendidas, gerando assim, uma nova configuração. FABRI et. al. (2004 e 2004(a)) propõem uma extensão, na qual ambas as partes do modelo possuem um grupo maior de atividades (a atividades propostas por FABRI, são delineadas na Figura 3.6) se comparada a Figura 2.6.

Dentro do contexto nacional, FERNANDES E TEIXEIRA (2004) propõem um modelo, denominado, neste trabalho, como classificatório, cujo objetivo é definir qual o escopo de fornecimento de produtos de uma fábrica de software. Em sua obra, os autores definem uma fábrica de software como: fábrica de projetos ampliada; fábrica de projetos de software; fábrica de projetos físicos; e fábrica de programas (código) (vide Figura 2.7).

Figura 2.7 – Modelo classificatório proposto para definição do escopo de fornecimento de uma fábrica de software (retirado de FERNANDES e TEIXEIRA (2004) p. 118)

A fábrica de projeto ampliada abrange o conceito de arquitetura da solução. A arquitetura da solução é um estágio anterior à conceituação do software, a qual se preocupa em projetar uma solução em que o software é somente um dos componentes. A arquitetura da solução pode conter, além do software, implantações de processos, definição de equipamentos, infra-estrutura de rede.

Já a fábrica de projetos de software abrange todo o ciclo de vida sistêmico para concepção do software (análise de sistema, projeto de software, construção, teste e implantação).

A fábrica de projetos físicos trata, exclusivamente, da concepção do software, abstraindo o modelo sistêmico, no qual o software está inserido.

Por fim, a fábrica de programas, considerada a menor das entidades, tem como objetivo desenvolver componentes de código para a montagem do software. Esta entidade não está preocupada com o contexto sistêmico nem com o projeto físico do software, ela apenas produz código segundo a especificação advinda do projeto físico. Resumindo, uma fábrica de programa possui como entrada uma especificação física de uma pequena parte do software e sua saída é caracterizada por um fragmento de código que irá compor o projeto físico.

Além do modelo classificatório, FERNANDES e TEIXEIRA (2004) apresentam uma estrutura genérica para a concepção de uma fábrica de programas (vide Figura 2.8). A arquitetura apresentada pelos autores possui as seguintes atividades:

Figura 2.8 – Modelo de fábrica de programas. Retirado de FERNANDES e TEIXEIRA (2004) página 32.

- **Recebimento e aceitação de artefatos:** Atividade que tem como objetivo verificar se os artefatos advindos da fase de projeto de software seguem o padrão de “completude” e “corretude” proposto pela fábrica de programas;
- **Planejamento e distribuição:** Nesta atividade, o projeto de software é particionado em ordens de serviços (OS), tais ordens são distribuídas às equipes para que sejam implementadas. Aspectos de custo e prazo para produção são estabelecidos nesta atividade;
- **Análise de Tarefa:** Nesta atividade, a ordem de serviço é padronizada seguindo os padrões pré-estabelecidos para codificação;
- **Execução da codificação:** Nesta atividade, as ordens de serviços são transformadas em unidades de software;

- Teste: As unidades de software são testadas, seguindo os procedimentos pré-estabelecidos. Se algum erro for detectado a atividade “análise da tarefa” é disparada, novamente;
- Homologação e liberação: Atividade que tem como objetivo verificar se as unidades de software possuem aderência aos requisitos representados pelo projeto de software.

As atividades pertinentes ao modelo proposto por FERNANDES e TEIXEIRA (2004) podem e devem ser customizadas pelas fábricas de programas. Na customização, alguns fatores devem ser considerados, entre eles destacam-se:

- O processo é padronizado, documentado, praticado e medido. Pessoas são treinadas para operá-lo;
- A capacidade de atendimento é planejada, na presença do cliente;
- Existência de um sistema de planejamento e controle de produção, rigor na alocação de recurso;
- Flexibilidade de ambiente, possibilidade de fabricar programas com mais de uma tecnologia;
- Recursos humanos com alto grau de flexibilidade, cada programador deve dominar um pequeno conjunto de linguagens de programação;
- Existência de controle da qualidade em relação a cumprimento das metas, inclusive controle estatísticos dos processos, para identificar maior incidência de desvios;
- O Controle de versões dos artefatos deve ser desenvolvido;

- Tempos de ciclo de produção da estrutura fabril para o desenvolvimento de software e o tempo de programação são padronizados, levando em conta o tipo de linguagem e a complexidade do programa a ser construído;
- As metas de desempenho, em termos de atendimento dos tempos padrões, produtividade dos programadores e nível de defeitos, são controladas;
- Existência de projetos de melhoria contínua do processo em função dos resultados das metas de desempenho;
- Fábrica operando em várias localidades (separação geográfica) utilizando processo padrão;
- O recebimento de especificações advindas do cliente e o envio de programas codificados são feitos por meio eletrônico.

Uma instanciação da arquitetura genérica proposta na Figura 2.8 também está presente na obra de FERNANDES e TEIXEIRA (2004). Nela, os autores estabelecem que o processo de produção de software de uma fábrica de programas possui as seguintes atividades:

- Análise da ordem de serviço⁵ (OS): Atividade cujo objetivo é verificar a “completude” e “corretude” dos artefatos de especificações presentes na OS;
- Construção de Código: Atividade cujo objetivo é desenvolver o código especificado na OS. Salienta-se que o processo de produção de construção de código deve seguir os padrões estabelecidos pelo processo;
- Planejamento do teste: Nesta atividade, os casos de testes⁶ são projetados;

⁵ Uma OS é um documento formal, presente no processo de produção de software, que reúne as especificações de uma unidade programável a ser desenvolvida pela fábrica de código.

- Preparação do teste: Atividade cujo objetivo é desenvolver os dados de teste (configuração física das entradas);
- Teste unitário: Atividade que tem como objetivo desenvolver o teste de acordo com os casos (atividade de planejamento) e com os dados gerados pela atividade de preparação.

FERNANDES e TEIXEIRA (2004), também, preocupam-se, ativamente, com a gestão do processo organizacional de uma fábrica de software⁷, este fato pode ser validado ao se efetuar uma análise da Figura 2.9, a qual apresenta uma visão consolidada das atividades instanciadas a partir do modelo proposto na Figura 2.8 concatenadas com o modelo de categorização por níveis (estratégico, tático e operacional) apresentado por TRINDADE (2006). Nota-se que o processo de produção de software da fábrica de programas encontra-se mapeado na visão processual e o modelo de categorização por níveis estabelece a visão gerencial da fábrica de programas.

Como complemento do processo de produção de software instanciado a partir da arquitetura genérica, FERNANDES e TEIXEIRA (2004) apresentam, em seu trabalho, o fluxo operacional de uma fábrica de programas. No fluxo, é possível verificar a presença das entidades: “negociação e análise”; “recebimento”; “planejamento” e “aceitação”; “planejamento e controle da produção” (PCP); “gestão da configuração” e “controle da qualidade”. A entidade “negociação e análise” é responsável por especificar o programa a ser desenvolvido, estimando a sua complexidade e por, emitir uma ordem de serviço para a fábrica de software. Já a entidade “recebimento” tem o objetivo de verificar se a ordem de serviço emitida “é uma ordem de serviço mesmo”. Se sim, os aspectos de planejamento e aceitação são desenvolvidos; caso alguma falha seja detectada na ordem de serviço (OS), a

⁶ “Casos de testes são especificações de entradas para o teste e de saídas esperadas do sistema, somadas a declaração do que está sendo testado” (SOMMERVILLE (2003) página. 377).

⁷ A gestão do processo organizacional também é uma preocupação de RUHE (1999), HUMPHREY (2001) e TRINDADE (2006).

fábrica de programas encaminha uma solicitação de complemento para o cliente. Ao atingir a entidade de “planejamento e aceitação”, a complexidade da OS é analisada e, se não existir divergências, a entidade de “planejamento e controle da produção” é disparada. A principal tarefa do PCP é dividir as ordens de serviço e distribuí-las para as células de produção e comunicar a entidade de planejamento sob tal divisão. A entidade “produção” codifica as ordens de serviço, realizando os testes unitários. A entidade “gestão da configuração” controla a liberação dos artefatos, inclusive o código produzido, estabelecendo assim o controle de versões. Por fim, a “gestão da qualidade” registra defeitos encontrados nas ordens de serviço e verifica se o código é aderente à especificação advinda da entidade cliente.

Figura 2.9 - Linha de Processo instanciada de uma fábrica de programa (Adaptado de FERNANDES e TEIXEIRA (2004))

Além do modelo classificatório, da arquitetura genérica utilizada na concepção de uma fábrica de programas, FERNANDES e TEIXEIRA (2004) classificam uma fábrica, de programas ou de projetos em dedicada e não dedicada. Na contextualização não dedicada, é possível encontrar o seguinte cenário:

- Aceitação de qualquer tipo de projeto e se especializa no domínio das aplicações;
- Organização de acordo com as fases do ciclo de produção com o estabelecimento de equipes alinhadas a uma determinada plataforma (por exemplo: *mainframe* e plataforma *distribuída*);
- Possibilidade de possuir fábricas dispersas, geograficamente;
- Possibilidade de possuir o *Project Management Office* (PMO) (Escritório de Gerenciamento de Projetos) para o suporte ao gerenciamento de projeto global;
- A fábrica de projetos não precisa, necessariamente, utilizar a fábrica de programa (característica presente somente na fábrica de projetos);
- Se existirem centros de produção operando de forma distribuída, a camada de gestão de projeto deve ser configurada localmente;
- Possibilidade de se configurar uma fábrica de programas centralizada para atender a todas as fábricas de projetos (característica presente somente na fábrica de projetos);
- Todas as fábricas (ou centros de produção) utilizam a mesma ferramenta de gestão.

A Figura 2.10 apresenta uma visão consolidada da fábrica de projetos não dedicada ou generalista. Em tal figura constata-se a presença da visão gerencial, relacionada com o modelo de caracterização por níveis apresentado por TRINDADE (2006), e da visão processual, que aglomera as atividades pertencentes ao processo de produção de software. Na parte inferior da Figura 2.10 é possível notar a delimitação da fábrica de projetos físicos (fábrica que engloba as atividades da fábrica de programas) e a fábrica de projeto de software, estrutura esta que se sobrepõe à fábrica de projetos físicos.

Figura 2.10 – Fábrica de projeto não dedicada e generalista (Adaptado de FERNANDES e TEIXEIRA (2004)).

Já na contextualização dedicada é mapeado o seguinte cenário⁸:

- Aceita qualquer tipo de projeto em determinadas plataformas, pré-estabelecidas e especializa-se no domínio da respectiva aplicação (característica presente somente na fábrica de projetos);
- Possibilidade de se ter centros de produção dispersos, geograficamente;

⁸ Alguns itens da contextualização dedicada são os mesmos da contextualização não dedicada, fato este que levou a repetição destes no texto.

- Produção organizada por cliente e por modelo negócio (característica presente somente na fábrica de projetos);
- Equipes organizadas por negócio e plataforma de desenvolvimento;
- Equipes de especificações devem ter conhecimento sobre várias técnicas de engenharia de software;
- Possibilidade de possuir o *Project Management Office* (PMO) (Escritório de Gerenciamento de Projetos) para o suporte ao gerenciamento de projeto global;
- Gestão de configuração é feita, considerando as características do produto e do cliente;
- Fábrica de projetos sente a necessidade de utilizar a fábrica de programa (característica presente somente na fábrica de projetos);
- Se existirem centros de produção operando de forma distribuída, a camada de gestão de projeto deve se configurada localmente;
- Todas as fábricas (ou centros de produção) utilizam a mesma ferramenta de gestão.

Por fim, a Figura 2.11 apresenta uma visão consolidada da fábrica de projetos dedicada. As considerações gerais efetuadas para o contexto não dedicado são as mesmas para o contexto dedicado.

Um outro aspecto não menos importante para a composição estrutural deste trabalho, também configurado por FERNANDES e TEIXEIRA, é a proposta de um fluxo operacional para uma fábrica de projetos de software. Neste fluxo, é possível perceber a presença das entidades: “negociação e análise”, “recebimento e

expedição”, “planejamento e aceitação”, “garantia da qualidade”, “PCP”, “produção” e, por fim, “garantia da qualidade e teste” (entidade esta relacionada à produção).

Figura 2.11 - Fábrica de projeto dedicada e não generalista (Adaptado de FERNANDES e TEIXEIRA (2004)).

A entidade “negociação e análise” elabora o projeto conceitual do software e emite uma ordem de serviço para a fábrica de projetos. Tal ordem é recebida pela entidade “recebimento e expedição”, esta entidade analisa a OS conforme os padrões estabelecidos, caso a ordem de serviço esteja respeitando os padrões pré-definidos, a entidade de “recebimento” encaminha tal artefato para a o “planejamento”. Nesta entidade são elaboradas as estimativas, o plano de projeto (na qual este é inspecionado pela entidade “garantia da qualidade”), os riscos são avaliados e um contrato entre cliente e fábrica de projeto de software é firmado. Após a assinatura de tal contrato, a entidade “PCP” assume a responsabilidade pelo fluxo de produção. O “PCP” tem como objetivo identificar e reservar os recursos para o desenvolvimento do projeto, elaborar as instruções de execução, criar uma “baseline” do projeto e distribuir a OS para a “produção”. A entidade responsável pela “produção” recebe e analisa a OS advinda da entidade “PCP”, aloca os recursos, inicia o projeto (especificando os requisitos) e aciona a fábrica de código para produção das unidades programáveis. De posse de tais unidades a entidade

“garantia da qualidade e teste” planeja o teste integrado, sistêmico e de aceitação, visto que o teste unitário foi desenvolvido pela fábrica de programas, validando, assim, o produto e encaminhando-o para o cliente. O cliente avalia a qualidade do software e, caso o mesmo não tenha mudanças, o produto é aprovado para implantação.

Delineado o panorama teórico sobre fábrica de software, este trabalho apresenta, na próxima seção um estudo de caso das fábricas nipônicas e americana SDC.

2.1 As Fábricas Estrangeiras

Este capítulo tem como objetivo apresentar um relato condensado de 4 fábricas japonesas (Hitachi, Toshiba, NEC e Fujitsu) e uma americana (SDC). Uma descrição detalhada sobre tais fábricas pode ser encontrada no anexo B deste trabalho. Salienta-se que as informações mapeadas nesta seção foram extraídas de CUSUMANO (1991). O autor deste trabalho considera tais casos como clássicos, devido à riqueza de detalhe utilizada por Cusumano em sua descrição, e a época na qual estes foram implementados.

2.1.1 Caracterização das Empresas

A SDC iniciou suas atividades na década de 1950 com o desenvolvimento de um sistema de controle e defesa do espaço aéreo para a força aérea americana. Um grande marco da empresa em questão foi a participação na missão Apollo. No início da década de 1960 a empresa possuía um corpo de colaboradores que chegava a 4000 pessoas. Em 1971 a SDC perdeu mercado, acarretando assim, uma redução no seu corpo de colaboradores, que nesta época chegou a 2000. Neste mesmo período a empresa em questão procurou se estabelecer como fábrica de software e teve uma recuperação nos anos seguintes. Por fim, em 1980, a SDC foi vendida para a *Burroughs Corporation* e esta, por sua vez, se fundiu com a *Sperry Corporation* em 1986, e formaram a *Unisys*. Este fato levou a SDC a se transformar

na *Paramax*, caracterizada como uma subsidiária da *Unisys*. Em 1995, a *Paramax* foi vendida para *Loral Corporation* e no ano seguinte a *Loral* vendeu-a para *Lockheed Martin*.

A Hitachi iniciou suas atividades em 1908 e se instalou como fábrica de software em 1969. Alguns fatores motivaram tal instalação, entre eles destaca-se a escassez de mão de obra qualificada. Em 1987, a empresa iniciou sua produção de sistemas de informação automatizados, pois até então a empresa produzia apenas ferramenta e aplicativos que eram hospedados em seus equipamentos, focando os seguintes segmentos de mercado: indústria; recursos humanos; contabilidade; e hospitais. Atualmente, a empresa desenvolve produtos que vão desde semicondutores, baterias, micro e mini-computadores a vários softwares nos mais variados domínios do conhecimento.

Fundada em 1939, a Toshiba se caracteriza, hoje (final da década de 1990 e início do século XXI), como uma das maiores produtoras de equipamentos eletrônicos do mundo. Em relação à sua produção de software, é importante destacar a fundação da fábrica ocorrida em no início da década de 1970. Nesta época, a empresa contava com 11.000 funcionários, na qual 1000 destes se caracterizavam como engenheiros. Com a fundação da fábrica, a Toshiba passou a entregar, na década de 1980, cerca de 50% de código reutilizável em seus produtos.

A NEC iniciou suas atividades no setor de tecnologia em 1899 e na década de 1970 se caracterizou como fábrica de software. Em 1980, a empresa possuía um corpo de profissionais que chegava a 18.000 colaboradores. Atualmente, a NEC possui uma ampla carteira de produtos e soluções focadas em consultoria de negócios, dentro desta ótica é possível destacar os setores de comunicação de dados e voz, mobilidade, multimídia, computação de alto desempenho, computação baseada em servidor e segurança. Na área de sistema de informação automatizado a empresa provê soluções para os domínios relacionados à educação, finanças (bancos), hotelaria e saúde.

A Fujitsu fundou sua fábrica de software em 1979 e em 1988 possuía 58.000 colaboradores, trabalhando em todas as partes do mundo. Até 1988, a empresa produzia software em vários domínios do conhecimento, destaque para o setor bancário, financeiro e hospitalar. Atualmente (início do século XXI), as aplicações para o gerenciamento de redes e banco de dados; o pacote integrado para a produção de software; ERP; CRM e comércio eletrônico caracterizam a carteira de produto de tal empresa.

Apresentada a caracterização das empresas que configuram esta seção, este trabalho apresentará na próxima seção os insumos e as atividades do processo de produção software das empresas, citadas nominalmente neste capítulo.

2.1.2 Insumos e Atividades do Processo de Produção de Software

Conforme apresentado por FERNANDES e TEIXEIRA (2004) (vide Figura 2.7), uma fábrica de software pode ser caracterizada como: fábrica de programas; fábrica de projetos físicos; fábrica de projetos software, e; fábrica de projetos ampliada. Salienta-se que os insumos do processo é que definem qual a característica da fábrica, por exemplo: se um fábrica de software recebe um projeto de software, fatiado ou dividido em ordens de serviços, neste caso uma ordem de serviço é caracterizada como um artefato e contém a especificação de uma parte ou um componente do software a ser implementado (insumos), esta se caracterizará como um fábrica de programas. Com base neste exemplo e a fim de prover uma maior aderência dos relatos efetuados por CUSUMANO (1991) à proposta efetuada por FERNANDES e TEIXEIRA (2004) ou autor deste trabalho caracteriza uma fábrica de software em ciclos de produção, denominando-os em:

- Longo: Uma fábrica de software de ciclo longo deve estabelecer as bases contratuais para o desenvolvimento do todo o projeto, realizar levantamento de requisitos, modelagem de negócio (NADLER e TUSHMAN (1999)), equalização do modelo de

negócio⁹, projeto de software, equalização do projeto de software, produção do código, teste modular ou unitário (os componentes são testados de forma isolada), teste de integração (os componentes são testados de forma integrada), teste de aceitação (o software é testado na presença do usuário ou cliente), gerenciamento de projetos e, por fim, a entrega do software (envolve questões como garantia do produto e treinamento dos usuários). As fábricas com este escopo devem possuir uma forte padronização dos artefatos (ou ativos de processo) pertencentes às atividades de modelagem de negócio e projeto de software. É importante ressaltar que tais atividades dependem, primordialmente, da mão de obra criativa dos envolvidos com o processo de produção;

- Médio: uma fábrica de software de ciclo médio, a modelagem de negócio não é caracterizada no processo fabril, ficando sob responsabilidade da fábrica apenas as atividades de equalização do modelo de negócio, projeto de software, produção de código, testes (modular, integração e aceitação), gerenciamento de projetos e a entrega do produto;
- Curto: Em uma fábrica de software de ciclo curto são contempladas as atividades de equalização dos artefatos do projeto de software (segundo Costa (2003) o objetivo desta atividade é verificar se os artefatos advindos da fase de projeto são padronizados e compreensíveis), produção de código, teste (de componente e integrado) e gerenciamento de projetos. A modelagem de negócio, o projeto de software, o teste de aceitação e a entrega são de responsabilidade da empresa contratante da fábrica de software de ciclo curto, sendo assim, é possível afirmar que a fábrica de ciclo curto passa a ser uma sub-contratada para a realização das atividades de produção de código e de testes.

⁹ A equalização do modelo de negócios tem como objetivo, verificar as especificações do modelo de negócio quanto à “completude”, “corretude” e consistência em relação aos requisitos do cliente, bem como assegurar seu entendimento, para que as mesmas possam ser colocadas em linha de produção (atingindo a atividade de projeto de software), mitigando os riscos de parada na produção e quebra na produtividade. A equalização da modelagem de negócio é uma atividade optativa e deve ser realizada quando a modelagem do negócio e o projeto de software são realizados por pessoas diferentes.

Ao analisar as informações apresentadas por CUSUMANO (1991), foi possível constatar que todas elas, excluindo a NEC que se caracteriza dentro do ciclo médio, são norteadas pelas atividades pertinentes ao ciclo longo.

Em relação às atividades do processo é possível verificar que:

- a SDC realiza as atividades de planejamento (gestão) de projeto, levantamento de requisitos, análise de sistemas, projeto de software, codificação, teste e entrega. Salienta-se que a SDC possui uma equipe de engenheiros trabalhando, internamente, nos clientes com o objetivo de percorrer as tarefas inerentes as atividades de análise de sistemas e projeto de software. Os artefatos gerados em tais atividades são repassados para a codificação, caracterizando assim, o conceito de contratação interna, delineado para as empresas brasileiras;
- a Hitachi realiza as atividades de modelagem de negócio, projeto de software, codificação, teste e entrega (ou implantação) do software (atividade esta que inclui o conceito de manutenção). Dentro de seu processo de produção a empresa, também, estabelece a idéia de contratação interna, na qual uma unidade organizacional desenvolve a modelagem de negócio e contrata a fábrica de software para a execução das atividades de projeto, codificação e teste de software;
- a Toshiba realiza as atividades de levantamento de requisitos, modelagem de negócio, projeto de software, codificação, teste, instalação e manutenção;
- a NEC realiza as atividades de planejamento (gestão de projetos), projeto básico, projeto detalhado (estas duas atividades caracterizam a atividade de projeto de software), implementação (codificação e teste unitário), integração (teste integrado) e inspeção;
- a Fujitsu realiza as atividades de planejamento (gestão), análise de sistemas, projeto da estrutura física do software, projeto modular (estas duas últimas

atividades citadas caracterizam a atividade de projeto de software), programação, testes - modular, de integração do sistema, operacional, de aceitação, efetuado no ambiente de cliente - entrega e manutenção.

2.1.3 Gestão Operacional, da Qualidade e de Configuração

Em relação à gestão operacional, no referencial bibliográfico proposto por CUSUMANO (1991) foi constatado que as fábricas realizam as atividades de planejamento, execução (percorrendo o processo em si) e controle.

Na atividade planejamento são definidos os recursos necessários para execução do projeto e cronograma.

Em relação às métricas de produtividade a Hitachi utiliza o número de programas desenvolvidos para um determinado módulo, e o número de módulos desenvolvidos para um determinado projeto. A Tabela B.2 provê uma noção de alguns números obtidos com o controle de projeto praticado pela empresa.

Ano	(%) Projetos Atrasados*	Ano	(%) Projetos Atrasados*
1970	72,4	1978	9,8
1971	56,6	1979	7,4
1972	36,3	1980	10,7
1973	13,0	1981	14,0
1974	6,9	1982	12,9
1975	9,8	1983	18,8
1976	16,8	1984	16,3
1977	16,1	1985	18,0

Tabela B.2 – Projetos de software atrasados. *Baseado no cronograma original. (Extraída de CUSUMANO (1991) página 187) (Tabela replicada neste local a fim de promover uma maior agilidade na leitura do texto).

Já a Toshiba utiliza como métrica o total de linhas de código entregues em um determinado período e o percentual de reuso do software praticado. A Tabela B.4 apresenta tais métricas agrupadas por ano.

Ano	Total de LCEA entregue (pessoa / mês)	Crescimento do Índice de produtividade (%)*	Varição do índice de produtividade (%)**	% de reuso***	Novo código (LCEA)****	Defeitos por 100 LCEA	Número de funcionários
Dados antes da criação da fábrica de software							
1972	1230						
1973	1390	13	+13				
1974	1370	11	-2				
1975	1210	-2	-12				
1976	1390	13	+15				
Dados depois da criação da fábrica de software							
1977	Dado não disponível						
1978	1684	37				7-20	1200
1979	1988	62	+18	13	1730		1500
1980	2072	68	+4	16	1740		1700
1981	2443	99	+18	29	1735		2050
1982	2595	110	+6	26	1920		2100
1983	2763	125	+7	41	1630		2150
1984	2931	138	+6	45	1612		2250
1985	3130	154	+7	48	1612	0,2-0,05	2300

Tabela B.4 – Produtividade da Fábrica de Software da Toshiba (Retirada de Cusumano (1991) página 240). *Crescimento do índice de produtividade em relação a 1972. **Calculo da variação efetuada sobre o ano anterior. ***Reuso: Entrega de software com até 20% de modificação. **** Calculo obtido através de LCEA (Linha de Código Entregues em Assembly) * [(100 - %reuso)/100] (Tabela replicada neste local a fim de promover uma maior agilidade na leitura do texto).

As medidas de controle utilizadas pela NEC são: data de início e término do projeto, diferença entre o orçado e realizado (custos), recursos humanos utilizados, quantidade de erros encontrados e tempo utilizado para correção dos erros. A Tabela B.6 provê uma noção de alguns números obtidos com o controle de projeto praticado pela empresa.

Divisão (Subsidiária)	Objetivo de qualidade a ser analisado pelo comitê	Resultado	
		Antes da criação comitê	Após a criação do comitê
Controle de transmissão.	Defeito médio	1,37 / klc	0,47 / klc
Sistema de informação automatizado operando em minicomputadores.	Defeito médio	0,35 / klc	0,25 / klc
Grandes aplicações.	Alteração nas especificações	Abaixo de 40 %	

Tabela B.6. – Resultados aferidos após o desenvolvimento do programa da qualidade da NEC (Adaptado de CUSUMANO (1991). Legenda: klc (1000 linha se código). (Tabela replicada neste local a fim de promover uma maior agilidade na leitura do texto).

A Fujitsu, por sua vez, tem como parâmetro de medida o número de linhas de código entregue por um programador em um determinado período. A quantidade de defeitos em um grupo de linhas de código também é mapeada pela empresa (vide Tabela B.9).

Ano	Defeitos/1000 linhas de código mantidas.	Método de Detecção de erros (estimativas)		
		Teste (%)	Revisão de código fonte (%)	Revisão de folha de projeto (%)
1977	0.19	85	15	
1978	0.13	80	15	5
1979	0.06	70	20	10
1980	0.05	60	25	15
1981	0.04	40	30	30
1982	0.02	30	30	40
1983	0.02	--	--	--
1984	0.02	--	--	--
1985	0.01	--	--	--

Tabela B.9 – Defeitos em linhas de código reportados pelos envolvidos com o processo e pelos usuários – Extraído de CUSUMANO (1991) página 352. (Tabela replicada neste local a fim de promover uma maior agilidade na leitura do texto).

Outra questão que deve ser destacada nesta seção é a qualidade do produto. As empresas estrangeiras se preocupam muito com este fato, isto pode ser comprovado ao analisar a Tabelas B.4 e B.6. Para todas as fábricas apresentadas nos relatos de CUSUMANO (1991), as atividades de teste e de manutenção corretiva servem como termômetro para medir se o produto desenvolvido atende os padrões da qualidade pré-estabelecidos. Algumas empresas aplicam o conceito de *rastreabilidade* delineado pela norma IEEE 830-1998¹⁰, destaque para NEC e Toshiba.

Um outro fato a ser destacado nesta seção é a qualidade do processo. Os atributos utilizados, pela maioria das empresas, para este fator, são os desvios ocorridos no processo de produção.

Por fim, questões sobre a gestão da configuração são detectadas na empresas nipônicas, porém CUSUMANO (1991) não as aprofundam.

¹⁰ Na época da concepção das fábricas de software estrangeiras a norma IEEE 830-1998 não tinha sido consolidada. Este fato mostra que as empresas apresentadas por CUSUMANO (1991) já desenvolviam as boas práticas encontradas na norma.

2.1.4 Ferramentas e Biblioteca de Componentes

Conforme apresentado na seção 2.1, uma fábrica de software deve possuir um conjunto de ferramentas para auxiliar a execução das atividades do processo de produção. As fábricas apresentadas por CUSUMANO (1991) não fogem desta premissa e possuem, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão gerencial e processual. Dentro desta ótica, é possível constatar que:

- a SDC possui, sob a luz da visão gerencial, uma ferramenta de gestão de projetos e controle da produtividade dos envolvidos com o processo. Já, sob a ótica processual, é possível encontrar nos relatos de CUSUMANO (1991) a ferramenta de armazenamento de código para reuso. Ressalta-se que as ferramentas corriqueiras, como compiladores e ambientes de desenvolvimentos, também, são utilizados pela empresa;
- a Hitachi possui, dentro da visão gerencial, a ferramenta CAPS (*Computer-Aided Software Production System*). Tal ferramenta é utilizada no planejamento e na gestão de um projeto de produção de software. A ferramenta CAPS é integrada a uma ferramenta CASE (*Computer-Aided Software Engineering*) na qual esta, por sua vez, é utilizada na automação do processo de produção de tal empresa;
- a Toshiba possui, junto a visão gerencial, a ferramenta de controle da qualidade do produto e de projetos. Além destas ferramentas é possível encontrar dentro da visão processual, uma ferramenta para definição e armazenamento de requisitos e outra para construção de programas;
- Já a NEC utiliza as seguintes ferramentas: gestão do projeto, da qualidade, produção do código, gestão de configuração, gerador de código e de caso de teste.

- por fim, Fujitsu possui, sob a ótica gerencial, uma ferramenta para planejamento e controle do projeto de software. Já sob a ótica processual, é possível encontrar um gerador de código fonte, uma ferramenta para automação da atividade de teste, um gerenciador de documentos e uma ferramenta para controle das solicitações de manutenção, cujo objetivo é capturar as melhorias de software propostas pelos clientes.

Enfim, ao analisar os relatos de CUSUMANO (1991), é possível constatar que todas as empresas estrangeiras praticam o conceito de reuso de componentes, fato que demanda a organização de uma estrutura de armazenamento para estes. Neste trabalho, tal estrutura é delineada como base de componentes.

2.1.5 Produtos Gerados e Forma de Organização do Processo

Nesta seção será apresentada a organização das fábricas em relação aos produtos gerados e a forma de organização do processo de produção. O primeiro mostra a orientação da fábrica quanto ao domínio de conhecimento (a fábrica foca um ou mais segmentos de mercado, por exemplo: desenvolvimento de software para o setor bancário e para o setor hospitalar), ao produto (a fábrica desenvolve um ou mais tipos produtos, por exemplo: para o setor o setor hospitalar a fábrica desenvolve somente o controle de retirada de remédios) e à tecnologia (a fábrica desenvolve para uma ou várias tecnologias – linguagens, plataformas, sistemas operacionais). Já o segundo busca respostas para a seguinte pergunta: O processo de produção das fábricas foi desenvolvido por meio e uma estrutura ou modelo pré-definido pela literatura, ou por outra área do conhecimento?

Em relação à orientação das fábricas é possível constatar que nenhuma delas é orientada a tecnologia, produtos e a domínio do conhecimento.

Já em relação à forma de organização do processo, constata-se que Toshiba utilizou a boas práticas relacionadas à produção de hardware (exemplo de outra área do conhecimento), além da experiência dos profissionais que nela atua, e as

aplicou para o desenvolvimento do processo de software. Uma descrição mais aprofundada sobre este fato pode ser verificada no anexo B, seção B.5.

Já a Fujitsu utilizou as seguintes estratégias, aliada a experiência dos profissionais que nela atua para organização de seu processo de produção:

- Busca por linguagens de alto nível que propiciassem uma alta produtividade no desenvolvimento do código fonte;
- Padronização das atividades do processo;
- Mecanização e automação;
- Treinamento dos envolvidos.

É perceptível que as estratégias utilizadas pela Fujitsu são caracterizadas como genéricas e, também, devem ter sido utilizadas por todas as fábricas que compõem este trabalho.

Por fim, CUSUMANO (1991) não relata quais foram às técnicas utilizadas pelas demais empresas.

2.2 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo

Baseado nas definições e nos casos apresentados neste capítulo, este trabalho aborda fábrica de software como uma organização estruturada, voltada para a produção do produto software, totalmente alicerçada na engenharia, e com forte caracterização pela organização do trabalho, pela capacidade de modularização de componentes e pela escalabilidade produtiva. Uma fábrica de software deve possuir um ambiente de gerenciamento de projetos; um processo

padronizado, definido e institucionalizado; políticas que garantam a qualidade do produto software; um conjunto de ferramentas para mecanizar o gerenciamento de projetos, processo e construção; técnicas para medir e estimar custo, prazo e tamanho de uma equipe para um determinado projeto; ambiente de teste definido e padronizado; foco em um segmento de mercado e; política de desenvolvimento de recursos humanos (FABRI et. al. (2004 e 2004(a)) e FABRI et. al. (2005)).

Além da visão sobre fábrica de software, apresentada no parágrafo anterior, é interessante que uma estrutura de produção de software com características fabris possua: uma unidade de produção de software; unidade de produção de componentes; modelo de estrutura organizacional; processo de produção de software definido e institucionalizado; processo de produção de componentes; medidas e estimativas; gerência de projeto de software; base de componentes; base histórica; métricas da qualidade e ferramentas.

Por fim, um grupo de pessoas qualificadas e atuantes nas visões: gerencial e processual é indispensável para uma estrutura de produção de software com características fabris possa se estabelecer e desenvolver software, com um alto grau de qualidade, para o mercado.

Capítulo 3 – O Modelo de Fábrica de Software Proposto pelo ELABSOFT

O Departamento de Engenharia de Produção da Escola Politécnica de Universidade de São Paulo mantém, ligado aos programas de pós-graduação (mestrado e doutorado), uma linha de pesquisa relacionada à gestão de tecnologia da informação. Nesta linha, encontra-se a de fábrica de software, cujo objetivo é desenvolver uma “fábrica modelo” para a produção de software. Tal pesquisa objetiva conceituar métodos, técnicas e ferramentas relativas ao contexto fabril para software, na forma mais ampla possível. O desenvolvimento de tais conceitos é caracterizado como contínuo e denominado ELABSOFT.

A principal razão para a existência do ELABSOFT é levantar, discutir, experimentar e gerir métodos, técnicas e conceitos ligados, direta e indiretamente, à aplicabilidade da engenharia de software em ambientes fabris para a produção de tal produto. O objetivo motriz dos pesquisadores é possibilitar o aumento da qualidade e produtividade do software por meio de processos otimizados, tanto de manufatura (fábrica em si), como de organização e gerenciamento da produção.

O modelo organizacional proposto para uma fábrica de software é direcionado por BASILI et. al. (1992) e composto de outras premissas tais como:

1. as de THORESON (1989): que apresenta uma organização de produção de software com características fabris definidas por uma estrutura funcional que contempla uma interface com o desenvolvedor e uma base histórica de projetos de software, provendo definições relacionadas a procedimento de produção (proximidade com a interface) e a de administração (próximos aos projetos). Além da interface e da base histórica de projetos, a autora apresenta duas linhas de atividades que são relacionadas de forma organizada, num fluxo não explícito mas, cronologicamente, evidente (vide Figura 3.1). Em uma das linhas é possível verificar as atividades de análise de sistemas, definição de interfaces, análises

estruturadas, linguagem de programação - produção (no caso relatado por THORESON (1989), a linguagem utilizada é Ada), teste automático e a atividade relacionada à gestão de projetos. Na linha próxima à interface com o desenvolvedor é possível perceber o catálogo de dados (documentos presentes na base histórica), técnicas para a formação de diagramas, projeto estruturado análise de código e documentação (manuais).

Figura 3.1 – Visão do aspecto produtivo de software (adaptada de THORESON (1989)).

2. as de VERRALL (1991): cujo trabalho faz parte do projeto Eureka, caracterizado como um consórcio que inclui 13 companhias de 5 países da Europa, tais companhias atuam nas seguintes áreas: manufatura de computadores, instituições de pesquisas, produção de ferramentas CASE e desenvolvimento de sistemas de informações gerenciais. A principal premissa do projeto Eureka é simples: garantir que uma fábrica de software seja composta de processos, regras, ferramentas, informações, pessoas e equipamentos (computadores). FERNSTROM (1991), ROCKWELL e GERA (1993) compartilham da idéia do autor citado neste item, e a visão arquitetônica de uma fábrica de software, proposta por eles, pode ser verificada por meio da Figura 3.2. Nesta figura, nota-se que o processo fabril é composto por regras, as quais são definidas pelos envolvidos com ambiente de desenvolvimento de software e estas originam padrões, algoritmos e métodos de desenvolvimento de software. As ferramentas e informações, armazenadas em uma estrutura computacional, suportam e automatizam o processo fabril. Baseado no trabalho de VERRALL (1991),

ROCKWELL e GERA (1993) propõem um modelo arquitetônico para fábrica de software com características de produção distribuída, nele existe a concepção da idéia do barramento de software (*software bus*). Barramento este que estipula regras de conexão de componentes. Na Figura 3.3 é possível verificar que o barramento conecta os componentes de serviço e os componentes de interface. Estes componentes podem ser desenvolvidos por diversas fábricas de software em locais diferentes e sua conexão acontece por meio de regras. A técnica de distribuição permite que as fábricas de software que utilizem este modelo possam compartilhar os componentes¹. Elaborando uma visão mais aprofundada do modelo, é possível notar que os autores classificam os componentes em Componentes de Serviço (CS) e Componentes de Interface (CI), facilitando, assim, o desenvolvimento de software multicamadas o que é, amplamente, polarizado no mercado.

Figura 3.2 - Modelo Eureka (inferido a partir dos relatos de ROCKWELL e GERA (1993))

Figura 3.3 - Barramento de produção de software (inferido a partir dos estudos de VERRAL (1991) e ROCKWELL e GERA (1993)).

¹ Técnicas sobre o compartilhamento de componentes e de desenvolvimento distribuído de software são aprofundadas por L'ERÁRIO (2006).

3.1 A Proposta Efetuada pelo ELABSOFT

De posse da base conceitual proposta por THORESON (1989), FERNSTROM (1991), VERRAL (1991), BASILI et. al. (1992), ROCKWELL e GERA (1993) e FERNANDES (2000), os pesquisadores do ELABSOFT desenvolveram um modelo organizacional para fábrica de software segmentado em visões (Figura 3.4):

- Visão metodológica²: aborda os mecanismos formais para análise e modelagem do negócio (ou análise de sistemas) nos quais o software irá operar (descrição do problema na visão do cliente). Nesta visão, também, é feita a modelagem conceitual dos componentes (possibilidade da utilização dos mesmos mecanismos) que irão formar o software no processo de manufatura;
- Visão tecnológica³: aborda questões técnicas para implementação das unidades programáveis (união de componentes para a composição de funcionalidades de um determinado software). Nesta visão os componentes de código, também, são implementados;
- Visão de negócio: visão que aborda questões de relacionamento entre cliente e a fábrica de software. Neste tipo de relacionamento, o cliente firma contrato com a fábrica de software, provê os requisitos para a modelagem e análise do negócio, no qual o software irá operar, e recebe as unidades programáveis para que estas sejam implementadas. Nesta visão é possível perceber todo o ciclo de produção do software, desde a coleta dos requisitos até a entrega;
- Visão de fabricação de componentes: abordagem interna, nesta visão é possível perceber a completude do ciclo de produção dos componentes, que fazem parte da composição das unidades programáveis. Ressalta-se que tal ciclo abrange o desenvolvimento de componentes classificados como componente de modelagem (ativos de processo: padrões, algoritmos e formulários).

² Visão ligada à fábrica de projetos de software apresentada na Figura 2.7.

³ Visão ligada à fábrica de programas apresentada na Figura 2.7.

Uma visão embrionária (ou inicial) do modelo proposto pelo ELABSOFT pode ser verificada na Figura 3.4.

É possível perceber a aderência da visão embrionária do modelo do ELABSOFT com a visão embrionária proposta por BASILI et. al. (1992) representada na Figura 2.5. A visão de fabricação de componentes assemelha-se à fábrica de componentes. A visão de negócio sobrepõe à organização baseada em projetos. O elo entre as visões metodológica e tecnológica se dá pelas atividades e tarefas relacionadas ao projeto de software.

Após o desenvolvimento da visão embrionária, os pesquisadores do ELABSOFT propuseram uma evolução do modelo organizacional, que possibilitou a inserção das trocas de informações entre os quadrantes, apresentados na Figura 3.4, e dos envolvidos⁴, mapeando assim, as responsabilidades dentro de cada quadrante. A evolução da visão embrionária do modelo organizacional para uma fábrica de software pode ser verificada na Figura 3.5.

Figura 3.4 – Visão embrionária do modelo organizacional para fábrica de software do ELABSOFT

⁴ Salienta-se que nos modelos de RUHE (1999), FERNSTROM et. al. (1992), CANTONE (1992), BASILI et. al. (1992), LI et. al. (2001) e HUMPHREY (2001), FENANDES e TEIXEIRA (2004), relatados na seção anterior, não foram especificados quais eram os envolvidos com o processo fabril de desenvolvimento de software.

Figura 3.5 – Evolução da visão embrionária do modelo organizacional para fábrica de software do ELABSOFT

O primeiro quadrante, representado na Figura 3.5, tem como objetivo estabelecer as negociações para o desenvolvimento de um novo software. Efetuadas as negociações, a modelagem do sistema de informação é desenvolvida pelo analista de sistema e de negócios. Se desejável for, tal quadrante pode ser focado sob o paradigma orientado a objetos, como proposto pelo Processo Unificado (o *Rational Unified Process* (RUP) – KRUCHTEN (2003)). Na orientação a objetos, a análise de sistemas assemelha-se à produção de casos de usos para especificação da dinâmica do negócio da empresa-cliente da fábrica de software. Porém, se desejável for, é possível estabelecer a ótica da orientação a processo em tal quadrante, aplicando a abordagem da análise essencial estruturada, derivando-se os seguintes artefatos: diagrama de contexto e lista de eventos. Tanto o levantamento de requisitos quanto a modelagem sistêmica utilizam os componentes de modelagem (ativos de processos, regras, *templates*, algoritmos, caracterizados aqui como componentes de negócio) armazenados na base (vide Figura 3.5). Ao utilizar os componentes, os envolvidos com o primeiro quadrante podem solicitar a alteração dos mesmos (ou propor novos), conforme sentirem necessidade (tal alteração (ou desenvolvimento) é realizada pelos envolvidos com o segundo

quadrante). Por fim, após a modelagem do sistema de informação, o engenheiro de software e o engenheiro de produção desenvolvem o projeto do software a ser fabricado. Salienta-se que as atividades relacionadas ao projeto do software estabelecem um elo entre os quadrantes 1 e 3.

O segundo quadrante tem como objetivo desenvolver e manter os componentes de modelagem. As atividades relacionadas a este quadrante são encarregadas do desenvolvimento (padronizações, métodos, processos, técnicas de gerenciamento) e do provimento de subsídios para a estruturação da base histórica de projetos (o documento que relata o histórico de um processo, também, é armazenado na base de componente). O desenvolvimento e a manutenção dos componentes de modelagem são estimulados pelos envolvidos com o primeiro quadrante. Uma vez desenvolvidos e aprovados pelos envolvidos com o ambiente organizacional, os componentes são armazenados na base e disponibilizados para serem utilizados pelo processo de produção de software.

O terceiro quadrante foca a questão da fabricação dos programas, estes desenvolvidos com base em especificações advindas das atividades pertinentes ao projeto de software, atividades estas que proveêm o elo entre a análise de negócio e a montagem do software. As atividades imersas em tal quadrante são disparadas pelos analistas de sistemas e engenheiros de software.

O quarto quadrante tem como objetivo desenvolver componentes de código que comporão as unidades programáveis, desenvolvidas pelas atividades imersas no quadrante 3. Encontram-se no quadrante 4, as atividades pertinentes ao projeto do componente, estas têm como objetivo desenvolver as especificações necessárias para que o componente seja codificado.

A evolução do modelo organizacional proposto traz, em si, algumas características propostas por BASILI et. al. (1992) e THORESON (1989). Em ambas, é perceptível o item base de componentes. Ressalta-se que THORESON (1989)

provê, em suas premissas, o “interfaceamento” do desenvolvedor com a base de componentes, fato também explicitado no modelo do ELABSOFT.

Com base no modelo organizacional de fábrica de software, proposto pelos pesquisadores do ELABSOFT, e visando atender as prerrogativas de se estabelecer um processo para o desenvolvimento de software, FABRI et. al. (2004, 2004a e 2004b) alicerçaram seus trabalhos.

O processo de produção de software com características fabris denominado por FABRI et. al. (2004, 2004a e 2004b), simplesmente, como processo fabril, define que uma fábrica de software possui duas unidades: produção de software e produção de componentes (Figura 3.6). A unidade de produção de componentes possui como ciclo de produção as seguintes atividades:

- **Projetar componentes:** Atividade que tem como objetivo modelar as funcionalidades, estrutura de dados e interface de um componente. Tal componente pode ser solicitado pela unidade de produção de software;
- **Construir componentes:** Atividade que tem como objetivo desenvolver componentes de código e de infra-estrutura (entende-se por componente de infra-estrutura, os ativos do processo, *templates*, formulários e algoritmos);
- **Integrar componentes:** Atividade que tem como objetivo verificar a qualidade funcional dos componentes. Em tal atividade são desenvolvidos os casos de teste, são preparados os dados de teste, os componentes são executados e, por fim, os dados resultados são comparados;
- **Armazenar componentes:** Esta atividade propõe que o administrador da base de componentes (“*data base component administrator*” (DBCA)) gerencie o armazenamento dos componentes, provendo permissões de acesso aos envolvidos com o modelo organizacional;

Figura 3.6 – Processo fabril ELABSOFT

- Distribuir componentes: Existem duas formas de distribuição de componentes, o componente pode ser vendido ou entregue aos envolvidos com o processo de produção para que unidades de software sejam manufaturadas.

FABRI et. al. (2004, 2004a e 2004b) salientam, explicitamente, que a unidade de produção de componentes alicerça a unidade de produção de software. Esta última possui as seguintes atividades:

- Análise de sistemas: Nesta atividade os eventos sistêmicos relacionados à empresa-cliente da fábrica de software são definidos;
- Projeto de software: Atividade que tem como objetivo desenvolver o projeto do software. Salienta-se que o projeto de software é composto por funcionalidades, estrutura de dados e interface, nesta atividade, também, é definido o tamanho do software⁵;
- Construção do software: Objetiva o desenvolvimento das unidades programáveis;
- Integração: Atividade que tem como objetivo verificar a qualidade funcional das unidades programáveis. Nesta atividade, são desenvolvidos os casos de teste, preparados os dados de teste, os componentes são executados e, por fim, os resultados obtidos são analisados;
- Implantação: Atividade que objetiva implantar o software, utilizando técnicas de configuração e aspectos relacionados ao treinamento. Na atividade implantar está embutido o conceito de manutenção do software;
- Revisão: Atividade que tem como objetivo trocar o componente de um software que está implantado, por um outro, que provenha maior otimização (por exemplo: relação a tempo de reposta).

⁵ Para a definição do tamanho do software são utilizados modelos e técnicas advindas da literatura. Para uma visão aprofundada de tais modelos consulte TRINDADE (1999).

Por meio da Figura 3.6, é possível verificar ainda, as atividades de gestão. Estas devem ocorrer em paralelo a cada atividade do processo de produção de software e de componentes. A concepção global da gestão também deve ser desenvolvida, fato este explicitado por FABRI et. al. (2004, 2004a e 2004b).

Os autores acima mencionados ressaltam, ainda, que as atividades analisar, projetar software, construir, integrar e implantar, da unidade de produção de software, são disparadas somente quando um novo produto é solicitado. Ao solicitar uma melhoria, é estimulada a execução de todas as atividades, exceto a análise e revisão. A atividade de revisão é estimulada quando for realizada uma solicitação por parte do cliente.

Já as atividades da unidade de produção de componentes são disparadas de duas formas: interna – as atividades da unidade de produção de software estimulam as atividades da unidade de produção de componentes; externa – possibilidade da venda de um determinado componente para um cliente externo.

Do ponto de vista da relação dinâmica entre as atividades propostas no processo fabril, os autores estabelecem que a unidade de produção de software une a idéia do processo incremental com a orientação ao reuso (SOMMERVILLE (2003)). Já a unidade de produção de componentes é permeada pelo modelo embrionário

O suporte para a proposta de FABRI et. al. (2004, 2004a e 2004b) encontra-se definido no modelo organizacional para fábrica de software do ELABSOFT apresentado na Figura 3.5 e no modelo proposto por ROCKWELL e GERA (1993). Neste último, é possível verificar que uma fábrica de software é composta por pessoas, processos e ferramentas.

TRINDADE (2006) apresenta, em seu trabalho, uma extensão do processo fabril proposto por FABRI et. al. (2004, 2004a e 2004b). Em seu estudo, o autor

afirma que o processo proposto por Fabri não percebe alguns pontos importantes para uma organização de produção de software com característica fabril. Fabri apenas foca a organização, manufatura e entrega, sem abrangência anterior, quando ocorre o contato de contratação de um projeto, análise deste como um negócio e como produto categorizável em ferramentas de software. Estes pontos incluem, no processo fabril, a atividade de negociação, atividade esta que é percorrida durante a análise e que contextualiza o produto de software e, ainda, observa a etapa de revisão como sendo parte integrante da negociação, num processo de pós-venda e pós-implementação, esta última, inclusive, tem parte de seu desenvolvimento na abordagem da negociação, fato este que diz respeito à entrega do produto.

Além de complementar o trabalho de FABRI et. al. (2004, 2004a e 2004b) com as atividades negociação e de modelagem, TRINDADE (2006) salienta que o processo de montagem pode ocorrer por meio de uma estrutura distribuída, semelhante à idéia proposta por VERRAL (1991) e ROCKWELL e GERA (1993), idéia esta denominada como barramento de software. Tal barramento deve ser entendido como uma estrutura conceitual, ou seja: uma representação que permita a visualização das comunicabilidades entre objetos que provêem serviços a outros objetos e destes também os receba, num processo de desenvolvimento dos elementos e acoplamentos destes por meio de suas interfaces. Ao que VERRAL (1991) denomina objetos e elementos, TRINDADE (2006) denomina componentes.

Analisando as considerações efetuadas por TRINDADE (2006), é perceptível que mais um nível de classificação no modelo proposto por FERNANDES e TEIXEIRA (2004) pode ser adicionado, denominado aqui como nível de contrato, sendo assim, é possível representar o modelo classificatório apresentado na Figura 2.7 na forma definida pela Figura 3.7⁶. O nível de contratação está presente em todas as fábricas: de projetos ampliados, de projetos de software, de projetos físicos e de programas, fato este que levou o autor deste trabalho a destacá-lo em negrito.

⁶ Ressalta-se que FERNANDES e TEIXEIRA (2004) citam o conceito de contratação em sua obra, porém não o insere no modelo classificatório.

Figura 3.7 – Extensão do modelo classificatório de FERNANDES e TEIXEIRA (2004), inferido a partir das considerações efetuadas por TRINDADE (2006).

As contribuições efetuadas por TRINDADE (2006), levaram os pesquisadores do ELABSOFT a proporem uma segunda geração do processo fabril. Nesta proposta é incorporada, também, a relação dos responsáveis com dinâmica de trabalho no processo, sugerindo assim, o estabelecimento de papéis e responsabilidades (FABRI et. al. (2006b)). O estabelecimento dos papéis ativos no processo respeitou o modelo organizacional para fábrica de software do ELABSOFT, apresentado na Figura 3.5. A segunda geração do processo fabril pode ser verificada por meio da Figura 3.8. Segundo TRINDADE (2006) e FABRI et. al. (2006b), ao analisar tal figura é possível perceber a presença de:

- **Administrador da base de componentes:** Representado pela sigla “DBCA”, é responsável por administrar os componentes armazenados na base. Tal profissional deve possuir conhecimentos relacionados à teoria de banco de dados, além da familiaridade com os dispositivos e ferramentas que gerenciam componentes. No processo atua diretamente sobre a base de componentes, e auxilia os demais no armazenamento e na recuperação dos componentes;
- **Analista de Negócio,** representado pela sigla “an”, responsável pela liderança e pela coordenação da modelagem processual relacionada ao negócio, delimitando assim, a organização de um problema em nível sistêmico, entende-se, neste caso, organização como empresa-cliente da fábrica de software. Sua principal

função, dentro do contexto do processo fabril, é perceber atores e agentes do negócio e como eles interagem, entre si, e com o ambiente (ambiente do cliente e problema do cliente). O principal artefato gerado pelo analista de negócio é a planta arquitetônica do negócio, na qual se detalham os fluxos de trabalho, de informações e suas entidades. Tal profissional deve possuir conhecimento do domínio do negócio e da tecnologia possível para sua automação, possuir capacidade de estabelecer relações bem definidas de comunicação e abstração. No processo apresentado pela Figura 3.8, tal analista atua na análise do negócio, tanto do software quanto de componente, e na implantação do software, por serem os dois pontos de contato com o cliente da fábrica, além de oferecer apoio às atividades de análise de sistemas;

- Analista de Sistema, representado pela sigla “as”, responsável pela liderança e pela identificação de requisitos do Sistema de Informação (SI) relativo ao problema percebido, delimitando-o e definindo suas funcionalidades, seus dados e sua interação com agentes e com o negócio. Tal analista apóia o planejamento e condução da revisão formal do modelo de sistema e do software, determinando as funcionalidades da solução proposta ao produto de software. Participa, ativamente, tanto dos projetos de software quanto de componentes, especificando o produto de software adequado para atender as demandas do SI modelado, estabelecendo suas fronteiras e orientando as atividades dos engenheiros e projetistas. Tal profissional deve possuir habilidade de comunicação e abstração, além de conhecimento dos domínios do negócio e da tecnologia. No processo, tal pessoa atua na análise do sistema e nos projetos de software e do componente, nesta última, estruturando todo o processo de produção, incluindo a atividade de teste. Tal profissional serve como elemento de apoio a diversas outras etapas, como a análise do negócio, a implantação e manutenção do software;
- Codificador, representado pela letra “c”, é responsável pelo desenvolvimento e pelos testes de componentes e software. Tal profissional deve possuir conhecimentos do aplicativo que está em desenvolvimento, além de familiaridade

com ferramentas próprias para criação, testes e automatizações de testes de código. No processo, atua na atividade de manufatura e no teste do componente; na montagem e na manutenção do software;

Figura 3.8 – Modelo dinâmico do processo fabril em uma fábrica de software (padrão estabelecido pelo ELABSOFT).

- Engenheiro de produção, representado pela sigla “ep”, responsável pela organização do processo de manufatura (programação da produção) e pelo controle, tanto da fabricação dos componentes, quanto da montagem do software. Gerencialmente, este profissional é encarregado da produtividade do

processo de produção como um todo. Verificar metas, alocar recursos, ajustar prioridades, coordenar interações e gerir mudanças são atribuições de tal profissional. Sua atividade requer experiência no desenvolvimento de software e habilidades para analisar e gerenciar configurações, riscos, estimativas e planejamentos, necessários a tomada de decisões dentro do foco do processo. Atua no processo sobre o âmbito da manufatura (software e componentes);

- Engenheiro de software, representado pela sigla “es”, responsável pela aplicação dos conceitos relacionados à engenharia de software e ao processo de fabricação do produto software, estabelecendo seu desenvolvimento organizado, por meio da introdução e avaliação de recursos técnicos de planejamento, a validação e avaliação dos projetos e de seus ciclos de produção. Lidera a coordenação de atividades técnicas, estabelecendo a arquitetura estrutural de projetos e de produtos, seus agrupamentos de elementos e suas interfaces. Tem a visão ampla do negócio, do problema, da solução. Experiência no domínio do problema, conhecer o aparato tecnológico para desenvolver a solução são características inerentes a tal profissional. Liderança para conduzir o esforço técnico e tomar decisões técnicas, no desenvolvimento do processo, orientando e buscando as metas de produtividade estabelecidas, focando resultados aceitáveis, são atributos indispensáveis a este tipo de profissional. Esta pessoa divide com o analista de sistema, a autoria do software e atua no processo, no projeto, na montagem e no teste de software e de componentes, além de apoiar as atividades de organização de software e de manufatura;
- Negociador, representado pela letra “n”, é responsável pelo contato direto com os clientes da fábrica de software, na negociação e na entrega do produto. Deve ter bons conhecimentos na arte da venda, perícia esta aliada ao conhecimento das características diferenciadas do produto de software e suas aplicabilidades nas mais diversas áreas empresariais;
- Projetista, representado pela letra “p”, é responsável pela coordenação da transformação do modelo de SI em modelo de Tecnologia da Informação (TI),

mais, precisamente, no modelo de software, assegurando que o produto responda a eventos com imediatismo, de acordo com seus requisitos de tratamento de dados, definidos por tabelas, índices, visões, restrições e parâmetros do banco de dados. É função do projetista criar pacotes de projeto, essenciais ao desenho técnico do produto de software e desenvolver tais desenhos, com seus memoriais de especificações detalhadas, inclusive para dar condições de teste aos componentes manufaturados e ao software montado. Tal profissional deve conhecer técnicas de análise e projetos de processo e banco de dados (ou classes, se trabalhar sob a orientação a objetos), arquitetura de sistemas e do ambiente, na qual será inserido o produto, além da linguagem com a qual este será desenvolvido. Em seu trabalho a experiência nas limitações tecnológicas, impostas pelo estado da arte, é requerida. No processo representado pela Figura 3.8, os projetistas atuam no processo de software e de componentes e apóiam a análise de negócio.

Os pesquisadores do ELABSOFT salientam, também, que além das funções relacionadas ao processo transformador, totalmente operacional, configurado no âmbito processual (vide Figura 2.3), existem outras funções alinhadas ao processo gestor (âmbito gerencial) demonstradas na Figura 3.9. Para configuração de tais funções tais pesquisadores levaram em consideração as observações feitas por, KWASNICK (1989), FELICIANO NETO e SHIMIZU (1996), CASSARO (1998), MELO (1999) e NATSUI (2002). A relação das funções administrativas da fábrica de software com visão dos níveis organizacionais definidos na Figura 2.3 pode ser verificada na Figura 3.9. Em tal figura é possível verificar a presença do:

- Executivo de negócios, representado pela sigla “↑N”, responsável pela visão estratégica da organização. Denota-se que tal executivo possui as atribuições concatenadas a definições participativas relacionadas a: recursos humanos, contábeis e financeiros e; abordagens comerciais e mercadológicas, necessárias à definição e contextualização do negócio. Na configuração operacional do negócio, este executivo cuida da definição de fatores críticos de sucesso e indicadores de desempenho voltados à mensuração de eficácia. Este papel

exige, do profissional, alto nível de conhecimento administrativo relacionado às áreas funcionais do marketing, do financeiro e dos recursos humanos. Características de empreendedorismo, de valorização dos ativos intelectuais da comunidade interna da organização, de perceber e converter competências e qualificações, de desenvolver capacidades de inovação organizacional, de contextualizar empreendimentos, situações e circunstâncias de interesse do negócio, de estabelecer relações de mercado e desenvolver matrizes estratégicas e táticas necessárias à organização, também fazem parte das responsabilidades de tal papel;

- Executivo de informação, representado pela sigla “↑I”, responsável pela visão do negócio relativa ao contexto sistêmico e de TI, tanto para projetar soluções para clientes quanto para estabelecer soluções internas de estruturação da informação e uso de aplicativos dentro da fábrica. Tal profissional propõe soluções com alto caráter de inovação tecnológica para serviços ou produtos. Promover o alicerce analítico para o desenvolvimento de negócios associados à organização e contemplar as estratégias relacionadas à informação, desenvolvendo novas tecnologias, e prospectar novos clientes e mercados de software são característica inerentes a tal papel. Cuidar das definições estratégicas relacionadas à tecnologia envolvida com a manufatura de software, como por exemplo: plataforma de suporte a gestão do negócio, é uma das funções do executivo de informação. Na realização da atividade executiva relacionada à informação, é aconselhável que se tenha amplo conhecimento sobre a empresa de software e sobre as tecnologias da informação, as quais envolvem ferramentas, métodos e soluções tecnológicas e de infra-estrutura, além de habilidades relacionadas a percepção e síntese de informações, sistemas e processos e soluções tecnológicas;

relacionados à administração de recursos, equipes, estimativas e riscos, são predominantes em um gerente geral. Liderança, bom relacionamento interpessoal e gerenciabilidade de tempos e métodos são atributos exigíveis para realização de tal função;

- Gerente de Operações, representado pela sigla “go”, é responsável por toda parte operacional do processo, controlando a execução e gerenciando o desenvolvimento do software como um todo. Há diversos subpapéis derivados de tal tipo de profissional em uma fábrica de software (em outros de tipos de empresas, não seriam estes):
 - Gerente de sistema computacional, responsável por manter o ambiente operacional de desenvolvimento ativo, gerenciando os aspectos inerentes a hardware e software. O desenvolvimento de cópias de segurança é umas das atividades de tal gerente. As exigências para o desempenho de tal papel são: bons conhecimentos de componentes, de hardware, de software, e de suas possíveis dependências, além de sistemas operacionais e de plataformas de desenvolvimento;
 - Gerente de configuração, responsável por disponibilizar e manter a infraestrutura geral, tecnológica necessária para o desenvolvimento dos projetos; oferecer todo o suporte necessário à fabricação de produtos, garantindo espaços adequados de trabalho para implementações, testes e manutenções são tarefas pertinentes a tal papel; e desenvolver o histórico de controle de mudança e correções de defeitos. Exigências para desempenho de tal papel são: conhecimento de princípios de gerenciamento de configuração e experiência em ferramentas de gestão de configuração;
 - Gerente de mudanças preocupa-se em organizar a documentação e a gerência da configuração e seu processo de modificação. Necessita ser,

devidamente, qualificado para executar estimativas de impactos, tanto em custo quanto em tempo, quando solicitações de mudanças ocorrem;

- Gerente de processos, responsável pelo processo de desenvolvimento, desde sua pré-configuração até sua pós-crítica, envolvendo as equipes de produção, a dinâmica de suas composições e os esforços (de avaliação e aprimoramento). Conhecimentos necessários: desenvolvimento de software e habilidades de comunicação;
 - Gerente da qualidade preocupa-se com os esforços para a realização de testes, decorrentes da capacidade de se garantir a qualidade da atividade, o que exige, fundamentalmente, domínio geral sobre os aspectos da engenharia de software. Habilidades interpessoais, de planejamento e de gerenciamento e conhecimento sobre o domínio, sistema ou aplicativo em teste, são conhecimentos necessários para desempenhar tal função;
 - Gerente de implantação está preocupado com a transição dos produtos para a comunidade de usuários, o que exige experiência com implantações de sistemas, comunicabilidade, coordenação de trabalhos e respeito a orientações administrativas.
- Analista de Negócio, representado pela sigla “an”, possui todas as responsabilidades, anteriormente, definidas (vide item que apresenta a descrição de tal papel), adicionadas à atuação no contexto gerencial:
 - Nível estratégico: aliado ao executivo de negócio, discute as determinações relativas ao negócio de software no mercado;
 - Nível tático: aliado ao diretor administrativo discute e define modelos gerenciais para projetos, segundo as perspectivas de atendimento a clientes e de atendimento aos quesitos da qualidade.

- Analista de Sistema, representado pela sigla “as”, possui todas as responsabilidades, anteriormente, definidas, adicionadas à atuação no contexto gerencial:
 - Nível tático: oferece suporte para a definição de metas de negócio, no que diz respeito aos modelos e planejamento com os quais os processos implantados irão operar;
 - Nível operacional: atua, juntamente, com os engenheiros e gerentes e auxilia a execução dos processos e nas análises dos resultados obtidos.
- Engenheiro de produção: representado pela sigla “ep”, possui todas as responsabilidades, anteriormente, definidas, adicionadas à atuação no contexto gerencial:
 - Nível estratégico: oferece suporte ao executivo de negócio, na concepção do modelo estratégico de produção de software a ser organizado na fábrica;
 - Nível tático: atua, juntamente, com engenheiros e gerentes definindo o modelo de engenharia (de produção) a ser adotado pela organização, além de contribuir com os modelos gerenciais em diversos setores da fábrica e com a gestão da qualidade do produto e do processo;
 - Nível operacional: atua com o engenheiro de software e gerentes na definição dos modelos de processo de manufatura e nos esquemas de avaliação e validação dos mesmos, além de oferecer suporte à produção.
- Engenheiro de Software, representado pela sigla “es”, possui todas as responsabilidades, anteriormente, definidas, adicionadas à atuação no contexto gerencial:

- Nível estratégico: oferece ao executivo de informações a concepção das estratégias de desenvolvimento a serem adotadas, para a estruturação organizacional tecnológica, para as decisões sobre plataformas gestoras e para as escolhas das técnicas de trabalho relativas ao tipo de fábrica de software desejável;
- Nível tático: comendo com o engenheiro de produção a gerência na definição do modelo de engenharia de software a ser adotado para a fábrica e na discussão da aplicabilidade das técnicas escolhidas, a serem implantadas para o trabalho, além de dar suporte ao RH, no que tange o pessoal especializado para este tipo de empresa;
- Nível operacional: atua, diretamente, aos demais elementos da gerência executiva a fim de garantir a aplicação da engenharia determinada e participar da modelagem e da aplicação dos processos de desenvolvimento, além de contribuir com definições relativas ao ferramental e suporte à produção (produção sob os aspectos tecnológicos).

3.2 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo

Este capítulo apresentou a evolução cronológica e conceitual do modelo de fábrica de software proposto pelos pesquisadores do ELABSOFT. É importante ressaltar que a proposta apresentada foi concebida a partir da interação direta dos participantes do grupo de pesquisa em Gestão de Tecnologia da Informação (GTI) do Departamento de Engenharia de Produção. As discussões para se implementar um modelo fabril para o desenvolvimento de software iniciaram-se no final no ano 2000. De 2004 a 2006 Fabri e Trindade, juntamente com os demais pesquisadores do grupo de pesquisa, iniciaram a divulgação do modelo no meio científico, divulgação esta transcrita para este trabalho.

Por fim, é possível afirmar que o modelo e o processo propostos pelo ELABSOFT não se encontram, totalmente, estabilizados e com certeza evoluções surgirão em um futuro próximo.

Capítulo 4 – Definição dos Conceitos de Meta-Processo e Reuso de Processo de Software

As pesquisas em tecnologia de processo de software tiveram início na década de 1980, representando uma importante melhoria na qualidade de tal produto (FIORINI (2001) e REIS (2002)). Nesta época, várias teorias, formalismos, métodos e ferramentas foram desenvolvidos, os quais direcionam os conceitos atuais relacionados à área de Engenharia de Software. Segundo PETERS e PEDRYCZ (2001), os objetivos relacionados à tecnologia de processos de software são:

- Definir ferramentas para estabelecer e descrever processos e acompanhar a sua execução, controlando, com certa rigidez, o desenvolvimento das atividades que ocorrem na produção do software, registrando os dados gerados com a execução do processo;
- Desenvolver e permitir a adoção de conceitos de melhoria da qualidade dos processos de produção de uma determinada organização, objetivando assim, monitorar e registrar os eventos ocorridos com a execução do processo para, posteriormente, estabelecer políticas relacionadas à otimização do processo em relação aos aspectos da qualidade;
- Estabelecer um controle seguro em relação à alocação e consumo de recursos durante a produção do software, coletando, assim, métricas de projeto e disponibilizando-as para consultas posteriores.

Focados nestes objetivos, inúmeros ambientes, arcabouços e ferramentas foram desenvolvidos com o propósito de estabelecer, executar, analisar e aperfeiçoar os processos de produção de software (DERNIAME et. al. (1999)). PRIETO-DÍAZ (1991) salienta que os pesquisadores, já na década de 90, verificavam a possibilidade de criar conjuntos de processos reutilizáveis que poderiam ser instanciados nos mais variados domínios do conhecimento,

independentemente, de sua complexidade. Dentro deste contexto foi concebida a idéia de meta-processo de software que, informalmente, pode ser interpretada como o conjunto de atividades bem definidas e utilizadas para a criação e organização de um processo. No início do ano 2000, as pesquisas nas áreas de meta-processo e de reuso de processo se tornaram mais evidenciadas e o objetivo deste capítulo é apresentar os conceitos inerentes a tais segmentos de pesquisa.

Para atingir o objetivo acima traçado, o capítulo foi dividido em 5 seções: A seção 4.1 apresenta as definições sobre processo de software. A evolução dos modelos de processo presentes na literatura é apresentada na seção 4.2. A seção 4.3 aborda os conceitos sobre meta-processo e reuso de processo de software. Na seção 4.4 são apresentados os requisitos necessários para a concepção de um meta-processo. Os modelos de reuso de processos são apresentados na seção 4.5. E, por fim, a seção 4.6 direciona as considerações do autor à frente deste capítulo.

4.1 Definições sobre Processo de Software

Segundo PAULA FILHO (2003), um processo é qualificado como um conjunto de atividades, parcialmente, ordenadas constituídas por métodos, práticas e transformações, usadas para atingir uma determinada meta. O processo pode ser visto como uma “receita” que é seguida por um projeto. Sendo assim, o projeto é considerado uma instanciiação do processo. É importante ressaltar que não se deve confundir um processo (por exemplo: uma receita de bolo) com o respectivo produto (o bolo) ou com a execução do processo por meio de um projeto (a confecção de um bolo por uma determinada cozinheira, em um determinado instante de tempo). Tomando como base o exemplo figurativo da receita do bolo, o autor afirma que um processo é composto por passos (atividades percorridas para a confecção do bolo), por pessoas (definido pelo papel da cozinheira), por insumos (produtos utilizados para a confecção do bolo, por exemplo: a farinha de trigo) e, por fim, o produto final produzido com a execução do processo (o bolo). Uma comparação entre os processos de produção de um bolo e de um software pode ser verificada na Tabela 4.1.

Processo	Culinário (o bolo)	Software
Atividades	Mexer, misturar, cozinhar, descansar, gratinar.	Planejar, acompanhar o projeto, codificar, testar
Papéis	Cozinheiro, Auxiliar de cozinha.	Coordenador de projeto, Analista, Programador, outros.
Insumos	Farinha de trigo, açúcar, baunilha.	Documento de especificação, código para testar, plano de projeto.
Produto de trabalho (intermediário ou final)	Cobertura do bolo (produto intermediário), bolo (produto final)	Plano de projeto (produto intermediário), software (produto final).

Tabela 4.1 – Comparação entre os processos de produção: culinário e de software.

PETERS e PEDRYCZ (2001) apresentam o processo como uma seqüência de atividades que produzem vários documentos, culminando em um produto satisfatório que possa ser utilizado pelo consumidor. No caso de um processo de software, o produto é classificado como um programa executável. Os autores salientam que um processo deve possuir uma retro-alimentação, característica que garante sua evolução. Para tais autores o processo pode ser visto de forma hierárquica, dividido em três níveis:

- Nível universal: definido durante as décadas de setenta e oitenta, cuja idéia era fornecer um modelo de processo de software que pudesse ser utilizado em qualquer projeto;
- Nível mundial: possibilidade de decompor o nível universal, tal decomposição seria incorporada em um projeto específico. Neste nível, as necessidades, os planos, os guias e os métodos são identificados para um determinado projeto;
- Nível atômico: Possibilidade de decompor o nível mundial (um processo específico) em atividades e tarefas.

SOMMERVILLE (2003) define processo como um conjunto de atividades e resultados associados que direcionam a produção do produto de software. O autor salienta que o processo de software é complexo e caracterizado como uma atividade intelectual, fato este que estabelece limitações em relação à sua automatização.

Para Sommerville não existe um processo universal para o desenvolvimento de software. Atualmente, as empresas que desenvolvem tal tipo de produto, estabelecem diferentes relações dinâmicas entre as atividades configurando, assim, o seu próprio processo de software.

PRESSMAN (2002) apresenta processo de software como um conjunto de passos previsíveis (ou arcabouço) utilizados para a construção do produto software. O processo define a abordagem dinâmica que é adotada quando o software é elaborado. Os participantes no processo de produção de software são: gerentes, engenheiros de software e usuários, estes últimos desempenhando papel fundamental no processo. Para a produção do software o processo necessita ser detalhado, fato este que depende das características do produto que a unidade produtora deseja construir, por exemplo: um processo pode ser apropriado para a criação de um software que controla a parte eletrônica de uma aeronave, enquanto um processo, inteiramente, diferente seria indicado para a construção de um software para controle de estoque de uma grande loja. Em sua definição sobre processo de software, o autor enumera uma série de produtos de trabalho, entre eles destacam-se: programas, documentos e dados produzidos em consequência da execução das atividades definidas no processo.

Em sua obra, PRESSMAN (2002) prevê que a caracterização de um processo é estabelecida por um pequeno número de atividades aplicáveis a um projeto de software, independente de seu tamanho e de sua complexidade; por um certo conjunto de tarefas, advindas da engenharia de software; por marcos de projetos; por produtos de trabalho e, por fim; por pontos de garantia da qualidade. Tarefas relacionadas a gestão de configuração e métricas de software devem estar em todo o processo (vide Figura 4.1).

Figura 4.1 – Universalidade de processo de software (retirado de PRESSMAN (2002) p. 22).

SLACK et. al. (2002) salientam que um processo de produção gera bens ou serviços ou um misto dos dois. O processo envolve um conjunto de recursos e insumos (*inputs*) usados para transformar algo em bens e serviços (*outputs*). Os autores apresentam alguns processos de produção nos mais variados domínios do conhecimento. Tais processos podem ser vistos na Tabela 4.2.

Operação	Recursos (input)	Processo	Bem ou Serviço (output)
Linha aérea	Avião. Pilotos e equipe de bordo. Equipe de terra. Passageiros e carga.	Transportar passageiros e carga pelo mundo.	Passageiro e carga transportado.
Loja de departamento	Produtos à venda. Equipe de vendas. Registros dos clientes.	Disponer os bens. Fornecer conselhos de compras. Vender os bens.	Consumidores e produtos juntos.
Gráfica	Impressoras e desenhistas. Prensas de impressão. Papel, tinta, etc.	Projeto gráfico. Impressão. Encadernação.	Material desenhado e impresso.
Polícia	Oficiais da polícia. Sistema de informação. Público (defensores da justiça e criminosos).	Prevenir crimes. Solucionar crimes. Prender criminosos.	Sociedade justa. Público com sentimento de segurança.
Fabrica de comida congelada	Comida fresca. Operadores. Equipamento de processamento de alimentos. Congeladores.	Preparação da comida. Congelamento da comida.	Comida congelada.
Software (entrada na tabela não está presente no texto original de SLACK (2002). Ela foi inserida a partir das considerações efetuadas pelos os autores da engenharia de software, nominalmente, citados neste capítulo).	Informações geradas pelo cliente do software. Engenheiro de software e programador. Linguagem de especificação do projeto de software. Linguagem de programação. Computador.	Projeto do software. Codificação do projeto. Teste do software.	Software implantado no cliente.

Tabela 4.2 – Algumas operações descritas como processo de *input*-transformação-*output*. (Adaptada de SLACK et. al. (2002)).

Por fim, HUMPHREY (1989), define que um processo de software deve possuir condições de entrada (artefatos, métodos e regras a serem seguidas na alimentação do processo), função de transformação, condições de saídas (regras a serem seguidas para que o produto seja gerado) e uma retro-alimentação que pode ser implementada em ambos os tipos de condições (entrada e saída).

Com base nas definições apresentadas, este trabalho encara o processo de produção de um produto categorizado como software, como um conjunto de atividades bem definidas e documentadas que quando aplicadas, sistematicamente, garantem um certo grau de qualidade na confecção do produto. Além do conjunto de atividades, o processo possui outros atributos como: matéria prima, mão de obra e recursos. Tais atributos são considerados os insumos do processo de produção. Salienta-se também que o processo deve possuir o conceito de retro-alimentação com o objetivo de garantir o caráter evolutivo do mesmo. A visão embrionária de processo de produção de software é apresentada na Figura 4.2.

Figura 4.2 - Visão embrionária de um processo (inferido a partir das considerações efetuadas por HUMPHREY (1989), PERTES e PEDRYCZ (2001), PRESSMAN (2002), SLACK et. al. (2002), SOMMERVILLE (2003) e PAULA FILHO (2003).

4.2 Evolução dos Modelos Procedimentais de Produção de Software

PETERS e PEDRYCZ (2001), PAULA FILHO (2003) e SOMMERVILLE (2003) ressaltam que a relação dinâmica entre as atividades do processo define o seu

modelo. Em suas obras, os autores enumeram vários modelos, entre eles é possível destacar:

- Caótico, também chamado por PAULA FILHO (2003) de "codifica-remenda": Neste tipo de modelo, o desenvolvedor do software parte apenas de uma especificação (ou nem isso) e codifica o software, e à medida que erros são descobertos as correções são realizadas. O autor deixa claro que tal modelo é o mais utilizado pelos desenvolvedores, considerado atraente devido a não exigência de conceitos relacionados à visão gerencial e organizacional presente nos processos de produção (vide modelo na Figura 4.3);

Figura 4.3 – Modelo caótico (adaptado de PAULA FILHO (2003))

- Modelo cascata: Neste modelo as atividades são executadas em seqüência, o que possibilita a demarcação de pontos de controle bem definidos quando o processo é gerenciado. Considerado um modelo rígido e burocrático, em que as atividades de requisitos, análise e projeto devem ser, extremamente, consistentes, pois erros no final do projeto não são permitidos, PETERS e PEDRYCZ (2001) relatam que o modelo possui uma baixa visibilidade para o cliente, que só recebe o produto no final do projeto (vide modelo na Figura 4.4);

Figura 4.4 – Modelo cascata. Adaptado de PAULA FILHO (2003).

- Modelo incremental (apresentado por SOMMERVILLE (2003) e PEDRYCZ (2001)): Modelo definido para minimizar as deficiências advindas do modelo cascata. Neste tipo de modelo, os requisitos do produto são identificados em um instante inicial e, em seguida, as demais atividades do processo são repetidas cada vez que há uma nova versão do produto ou quando um novo módulo do software é identificado. Na relação dinâmica incremental, a confecção do produto é particionada, cada parte percorre as atividades e são apresentadas ao usuário no momento da implantação (vide o modelo na Figura 4.5);

Figura 4.5 – Modelo incremental inferido a partir dos relatos feitos por SOMMERVILLE (2003) e PETERS e PEDRYCZ (2001).

- **Modelo Espiral:** Neste modelo o produto é desenvolvido em uma série de iterações. BOEHM (1988) relata que cada iteração define uma volta no modelo espiral. SOMMERVILLE (2003) salienta que tal modelo combina as características positivas da gerência de documentos associados às fases do ciclo presente no modelo cascata com a sobreposição pertinente no âmbito espiral. O modelo em questão parte do princípio de que a forma do desenvolvimento de software não pode ser, completamente, determinada de antemão. Uma visão ilustrativa do modelo espiral pode ser verificada na Figura 4.6;

Figura 4.6 – Modelo espiral (retirado de PAULA FILHO (2003) p.19).

- Modelo de prototipagem: Segundo PRESSMAN (2002), este modelo se inicia com a definição dos requisitos, contemplando os objetivos gerais do software por parte do cliente e do desenvolvedor e estabelecendo as áreas mais complexas para o desenvolvimento de novas definições. Um “projeto rápido” é então realizado. O projeto concentra-se na representação daqueles aspectos do software que vão ficar visíveis ao cliente (exemplo: padrões de entradas e saídas de dados). O projeto rápido é estabelecido via protótipo, o protótipo é avaliado pelo cliente e utilizado no refinamento das áreas complexas do produto estabelecidas no início do levantamento de requisitos. Interações ocorrem à medida que o protótipo é ajustado para satisfazer os requisitos do cliente, e estas permitem ao desenvolvedor entender melhor o que precisa ser construído. Pode ser que em um determinado momento o protótipo passa a ser visto como produto¹. Uma visão gráfica do modelo prototipagem é espelhada na Figura 4.7.

SOMMERVILLE (2003) apresenta o modelo de prototipagem relatado por PRESSMAN (2002) com a denominação evolucionária. O primeiro autor ressalta que:

¹ Salienta-se que o protótipo tem como finalidade demonstrar os aspectos importantes do produto final. Em alguns casos ele, também, é descartado não se transformando em produto.

“o modelo evolucionário tem como base a idéia de desenvolver um implementação inicial, expor o resultado ao comentário do cliente e fazer seu aprimoramento por meio de muitas versões, até que um sistema adequado tenha sido desenvolvido. Em vez de se ter as atividades de especificação, desenvolvimento e validação em separado, todo esse trabalho é realizado, concorrentemente, com um rápido feedback por meio dessas atividades” (SOMMERVILLE (2003) p. 39).

Figura 4.7 – Modelo prototipagem (extraído de PRESSMAN (2002) p. 29).

SOMMERVILLE (2003) relata, ainda, que o modelo evolucionário possui duas visões:

1. Desenvolvimento exploratório: Trabalhar com cliente e explorar os requisitos do produto e entregar o software pronto como produto final.
 2. Fazer protótipos descartáveis: Compreender os requisitos do produto, a partir disto, desenvolver uma melhor definição de requisitos complexos desenvolvendo um protótipo, descartar o protótipo utilizado na compreensão destes requisitos.
- Desenvolvimento orientado a reuso: Aborda a ampla base de componentes de software reutilizável. Partindo dos requisitos do software, o projetista verifica quais os componentes que aderem a tais requisitos, se tais componentes não existem, eles são desenvolvidos. Tais componentes são encapsulados e reutilizáveis em um outro projeto (SOMMERVILLE (2003)). O modelo orientado a

reuso pode ser utilizado em conjunto com outros modelos, tais como incremental e o espiral.

Além dos modelos de processo apresentados por PAULA FILHO (2003), SOMMERVILLE (2003), PRESSMAN (2002), PETERS e PEDRYCZ (2001), é possível encontrar, na literatura, uma estrutura base para a definição de processo. Segundo PAULA FILHO (2003 páginas 29, 30, 31 e 32) nesta estrutura é possível verificar que o processo é composto por:

- Passo: “Divisão formal de um processo, com pré-requisitos, entradas, critérios de aprovação e resultados definidos”;
- Fase: “Divisão maior de um processo, para fins gerenciais, que corresponde aos pontos principais de aceitação por parte do cliente”;
- Iteração: “Passo constituinte de uma fase, no qual se atinge um conjunto bem definido de metas parciais de um projeto”;
- Fluxo: “Subprocesso caracterizado por um tema técnico”;
- Etapa: “Passo constituinte de um fluxo”;
- Atividade: “Termo genérico para unidades de trabalho executadas em um passo”.

A nomenclatura apresentada por PAULA FILHO (2003) é utilizada para a definição das unidades de trabalho que compõem o processo de produção de software por ele apresentado. Tal nomenclatura assemelha-se a do Processo Unificado da *Rational* (RUP) (KRUCHTEN (2003)), tanto no âmbito das fases (subprocessos gerenciais) quanto nos fluxos (supprocesso técnicos). O autor relata que uma fase é composta por uma ou mais iterações. Um fluxo é particionado em uma ou mais etapas.

Por fim, o autor apresenta um diagrama relatando a composição estrutural de um processo (vide Figura 4.8).

Figura 4.8 – Composição de um processo de produção de software (retirado de PAULA FILHO (2003) p. 28). O diagrama deve ser lido da seguinte forma: Um processo agrega uma ou mais fases e um ou mais fluxos, a(s) fase(s) agrega(m) uma ou mais iterações, o(s) fluxo(s) agrega(m) uma ou mais etapas. As etapas e as iterações são classificadas como passos.

Já DERNIAME et. al. (1999) apresentam, em seu trabalho, uma estrutura básica para a modelagem do processo, estrutura esta composta dos seguintes elementos:

- **Atividade:** definida como um passo de um processo. As atividades, geralmente, modificam os artefatos; elas incorporam e melhoram procedimentos, regras e políticas;
- **Conjunto de artefatos** a serem desenvolvidos, entregues e mantidos pelos projetos. Os artefatos também podem ser chamados de produtos;
- **Recursos:** Elementos necessários para que uma atividade seja realizada. No processo de produção de software, os recursos são classificados como *agentes* (recursos humanos) e ferramentas, estruturas computadorizadas que, tradicionalmente, devem ser utilizadas no desenvolvimento de software.

A relação entre os elementos propostos por DERNIAME et. al. (1999) pode ser verificada por meio da Figura 4.9. A leitura da figura deve ser feita da seguinte

forma: Uma atividade possui regras, ferramentas e produtos (artefatos do processo). A execução da menor unidade de uma atividade, a regra, é de responsabilidade de um agente, denominado na figura por executor.

Figura 4.9 - Composição de um processo de produção de software (adaptado de DERNIAME et. al. (1999)).

De posse das idéias apresentadas por PAULA FILHO (2003) e DERNIAME et. al. (1999) e com o objetivo de adequar a composição arquitetônica de um processo estabelecida por eles, este trabalho verifica que um processo de produção de software possui:

- Atividades: Aglomera um conjunto de tarefas que são executadas para a produção de artefatos de trabalho;
- Fluxo: Relaciona as atividade e as tarefas;
- Tarefas: Menor unidade de uma atividade, possui entradas, processo transformador e saída;
- Artefatos: Produtos de trabalho gerados pelas tarefas e, conseqüentemente, pelas atividades;

- Iteração: Atividade que tem como objetivo verificar se as definições organizacionais relacionadas à gerência do processo estão sendo atingidas;
- Ferramentas: Infra-estrutura, geralmente, computadorizada para automatizar as atividades na confecção dos produtos (artefatos).

Finalizando esta seção, um diagrama, representando a composição arquitetônica de um processo, proposta neste trabalho, é apresentado na Figura 4.10.

Figura 4.10 - Composição de um processo de produção de software (visão deste trabalho) O diagrama deve ser lido da seguinte forma: Um processo agrega uma ou mais atividades e uma ou mais pessoas; a(s) atividade(s) agrega(m) uma ou mais tarefa(s), zero ou mais ferramentas e um ou mais fluxo(s); a(s) tarefa(s) agrega(m) um ou mais artefatos. As iterações são classificadas como atividades. O processo também está relacionado com o modelo de processo e com os padrões, normas e modelos da qualidade.

Apresentado o levantamento bibliográfico sobre processo de software, a próxima seção irá apresentar os conceitos sobre meta-processo e reuso de processo de software.

4.3 Meta-processo e Reuso de Processo de Produção de Software

Segundo DERNIAME et. al. (1999), a teoria de meta-processo estabelece procedimentos para a definição, configuração, instanciação e execução de processo de software. Entre tais procedimentos é possível destacar:

- o de modelagem dinâmica das atividades do processo: Procedimento que retrata a representação global da estrutura de um processo de software. Exemplos: cascata, espiral, incremental e outros. O propósito deste procedimento é endereçar metodologias e modelos amplos que definem a relação dinâmica entre as atividades do processo a ser executado;
- o de modelagem genérica de processo: Procedimento que parte de uma configuração abstrata do processo, configuração esta que pode ser utilizada em vários projetos similares, desde que estes possuam características e propriedades semelhantes, e organiza um processo de produção de software a ser aplicado a um projeto. É importante salientar que para instanciar este procedimento e produzir uma representação de um processo executável é necessário definir os aspectos da modelagem dinâmica apresentada no tópico anterior;
- procedimento de modelagem de processo customizado: Neste caso o processo é detalhado e, usualmente, derivado da modelagem genérica. Na aplicação de tal modelagem, um exame das características do cliente é desenvolvido e o processo é customizado. DERNIAME et. al. (1999) salientam que existem vários níveis de customização relacionados ao aumento de detalhe e formalidade do processo, entre tais níveis é possível destacar o:
 - de processo: o processo é, inteiramente, modelado a partir de um processo genérico;
 - de customização e de padronização organizacional: ajustar um processo já existente em uma organização *A* para uma organização *B*;
 - de nível de customização de projeto: ajustar um processo aplicado ao projeto *C* e aplicá-lo ao projeto *D*;

Além dos procedimentos, utilizados definição, configuração, instanciação e execução de processo de software, a teoria de meta-processo propõe que um processo de software deva ser configurado em perspectivas. Entre elas DERNIAME et. al. (1999) e SOMMERVILLE (2003) destacam a de:

- Fluxo de trabalho: Perspectiva que foca o tipo, estrutura e propriedades das atividades e suas relações. DERNIAME et. al. (1999), destacam que tal perspectiva é relevante sob a ótica de gerenciamento de projetos, principalmente, na proposta de cronograma;
- Fluxo de dados: Descreve o tipo, a estrutura e as propriedades dos artefatos gerados pelo processo de produção de software. DERNIAME et. al. (1999), salientam que tal perspectiva interessa, principalmente, ao usuário, pois ela garante o entendimento da documentação que será entregue com o produto (o software);
- Recursos (geralmente ferramental): Descreve os recursos necessários a serem fornecidos para a execução do processo de software. Perspectiva esta relevante para o âmbito gerencial;
- *Skills* (habilidades): Descreve um conjunto peculiar de habilidades e responsabilidades necessárias para execução do processo. Esta perspectiva é relevante para o provimento da garantia da qualidade do processo e, consecutivamente, do produto.

Um outro fator importante inserido no contexto teórico tratado nesta seção é a forma de desenvolvimento (criação) de processo. DERNIAME et. al. (1999), REIS (2002) e FRANCH e RIBÓ (2002) apresentam três vertentes para tal forma:

- Vertente abstrata: Foca os elementos ou moldes para a solução de um problema comum. Pode-se comparar uma forma abstrata à definição de *patterns* ou *templates* (padrões). A abstração é classificada como uma estrutura de alto nível,

e sua principal função é fornecer elementos reguladores para a caracterização das atividades e interações entre cargos (gerentes, analistas e desenvolvedores) e ferramentas dentro de um processo específico;

- Vertente instanciada: Aborda as questões relacionadas à execução dos processos. Nesta forma, os processos são instanciados e possuem, em sua concepção genética, os elementos ou moldes salientados na vertente abstrata. É necessário definir os objetivos e restrições específicas inerentes ao processo, envolvendo agentes, cronograma, orçamento e recursos que um processo necessita para aderir a um projeto;
- Vertente de execução ou executada: Armazena todos os dados históricos gerados com a execução de um processo, informações como eventos, desvios, medidas, tempo de execução e modificações são estabelecidas a partir dela.

Analizadas as três vertentes, é possível perceber a relação de continuidade entre elas, por exemplo: a partir de uma abstração é possível instanciar um processo e, posteriormente, percorrê-lo (aplicação do processo em um projeto).

Além da relação de continuidade, REIS (2002) reforça que, na vertente de execução, devem ser consideradas questões como: interatividade e coordenação de múltiplos atores envolvidos no processo; interações destes atores com as ferramentas que automatizam o processo e; infra-estrutura para que os registros históricos do processo sejam armazenados e disponibilizados. Sendo, assim, a execução do processo é a camada que adere diretamente ao projeto de software. O nível de detalhe da vertente de execução deve relacionar-se, diretamente, com a máquina de processo, fator este de importante para o sucesso da institucionalização do processo.

SCHAEFER et. al. (1999) afirmam que a máquina de processo pode ser classificada como um software com o objetivo de auxiliar na comunicação e coordenação das atividades realizadas pelos envolvidos no processo. Já REIS

(2002) apresenta que a máquina de processo deve possuir características auto-executáveis, concebidas a partir da formalização e interação de elementos que compõem o processo, cujo objetivo é gerar, automaticamente, novas atividades do processo. Nota-se que a teoria proposta por REIS (2002), baseia-se na engenharia de software e processo de software sensível ao contexto. Este trabalho está tendendo a interpretar o termo máquina de processo segundo as considerações propostas por SCHAEFER et. al. (1999).

Uma outra linha de pesquisa relacionada às idéias de criação, instanciação e organização de processo é o conceito de processo reutilizável (ou reuso de processo). FIORINI (2001 p. 22) define que: “A *reutilização de processo* é o uso de um modelo de processo, na criação de outro processo”. Analisando a definição de meta-processo apresentada por DERNIAME et. al. (1999), no início desta seção, e de posse da afirmação efetuada pela autora, é possível verificar a relação entre as duas teorias citadas, nominalmente, no início deste parágrafo. Em sua obra, a autora remete o leitor a interpretar o conceito de reutilização como um *framework*².

Existem vários *frameworks* que aplicam as vertentes, os procedimentos e as perspectivas apresentadas neste trabalho, a fim de promover o conceito de reutilização de processos de software, entre eles destacam-se os trabalhos de JØRGENSEN e CARLSEN (2001), FRANCH e RIBÓ (2002) e REIS (2002). Em seu *framework*, REIS (2002) parte de um processo genérico e, por meio do conceito de adaptação (*tailoring*), modela um processo (concebendo a máquina de processo) com a capacidade de ser executado. Após a execução do processo, o mesmo é avaliado e melhorias são incorporadas no processo genérico. O *framework* proposto pelo autor é caracterizado como cíclico e demonstrado na Figura 4.11.

Desenvolvendo uma análise aprofundada do *framework* de reutilização proposto por REIS (2002), é possível concluir que a vertente abstrata adere ao termo “processo genérico” (Figura 4.11), a vertente instanciada adere à concepção da

² Em uma definição adaptada de FAYAD (1999), pode-se dizer que um *framework* é um esqueleto de uma aplicação ou processo, que será customizado pelo engenheiro de processo.

máquina da processo, concepção esta direcionada por meio da instanciação do processo genérico. A vertente de execução adere ao conceito de processo específico aplicado ao projeto. Por fim, uma quarta vertente pode ser inferida a partir da proposta de REIS (2002), vertente esta classificada como de avaliação, cuja responsabilidade é analisar os resultados gerados com a execução do processo, incorporando os aspectos positivos e excluindo práticas negativas do processo genérico.

Figura 4.11 – Cenário geral para a reutilização de processos de software (Adaptado de REIS (2002)).

FRANCH e RIBÓ (2002) propõem um *framework* semelhante ao de REIS (2002), denominado “*framework* modular de reuso” (vide Figura 4.12). Nesta proposta, a definição de um processo de software é confeccionada em 3 camadas: genérica, executável e executada. Na primeira, um processo genérico é definido. Após esta definição, um mecanismo de adaptação gera uma versão executável do processo (segunda camada). De posse desta versão, o processo é aplicado a um determinado projeto, atingindo, assim, a terceira camada. FRANCH e RIBÓ (2002) posicionam o seu *framework* sob duas óticas: *top-down* (de cima para baixo) e *bottom-up* (de baixo para cima). A *top-down* parte da primeira camada e atinge a terceira gerando, assim, dados reais sobre a execução do processo. A *bottom-up* se

apropriada de tais dados com o objetivo de incorporar os aspectos positivos e excluir práticas negativas ao processo genérico.

O *framework* de FRANCH e RIBÓ (2002) está alinhado às vertentes estabelecidas por DERNIAME et. al. (1999), incluindo a vertente de avaliação inferida a partir dos estudos propostos por REIS (2002). Esta afirmação pode ser validada por meio da Figura 4.12, na qual é perceptível que um processo genérico e composto a partir de processos modelos (relação com os procedimentos de modelagem dinâmica das atividades do processo³) direcionando, assim, a vertente abstrata. A partir do processo abstrato (utiliza-se o termo *generalista* na Figura 4.12) um processo executável é instanciado, utilizando um mecanismo de adaptação e estabelecendo, assim, a vertente instanciada. A vertente executada é concebida quando o processo executado sobre um projeto. Por fim, a vertente de avaliação é configurada sobre a linha *bottom-up* da Figura 4.12.

Figura 4.12 - *Framework* modular de reuso (adaptada de FRANCH e RIBÓ (2002)).

³ Procedimento que retrata a representação global de um processo de software, exemplos: modelos cascata, espiral, incremental e outros.

Por fim, DERNIAME et. al. (1999) deixam claro que os modelos de processo, os procedimentos, as perspectivas e os mecanismos apresentados nesta seção são elementos que auxiliam a estruturação e organização de processo de produção de software. Já AVRILLIONIS et. al. (1999) destacam que somente os modelos, os procedimentos, as perspectivas e os mecanismos não são suficientes para a estruturação de um meta-processo, sendo assim, eles enumeram um conjunto de requisitos para sua definição de um meta-processo, os quais são detalhados a seguir.

4.4 – Requisitos para a Definição de um Meta-processo

Segundo AVRILLIONIS et. al. (1999) os requisitos essenciais para a concepção de um meta-processo são:

1. Simplicidade: As atividades dos processos reais, aqueles que estão em constante contato com o ser humano são, demasiadamente, complexas. Dentro do âmbito conceitual de meta-processo é possível que suas atividades venham a ser mais complexas se comparadas ao processo que ele define. Este fato inspira cuidado na proposição de um meta-processo, com isto, o autor de uma estrutura dinâmica utilizada para a criação de processo de software deve sempre pensar de forma simples e, fortemente, estruturada.
2. Generalidade: O meta-processo deve ser genérico, o que possibilita a sua adaptação e customização para um domínio específico, ou ambiente no qual será implementado.
3. Extensividade: O meta-processo deve possuir características abertas, capaz de envolver, em suas “meta-atividades”, experiências ou conhecimentos desenvolvidos com a execução dos processos criados a partir dele.
4. Flexibilidade: O meta-processo deve ser concebido de tal forma que o desenvolvedor do processo seja capaz de adaptar suas práticas em um projeto,

assegurando, assim, uma otimização das atividades e tarefas relacionadas a este nível. O preenchimento estrutural de um meta-processo deve focar diferentes paradigmas (por exemplo: possibilidade de criar um processo de produção de software alinhado ao paradigma orientado a processos ou a objetos). O caráter adaptativo envolvendo mudanças corretivas, questões relacionadas à melhoria de suas meta-atividades, possibilitando refinamentos em sua estrutura, são aspectos de flexibilidade que devem ser incorporados em um meta-processo.

5. Robustez: O meta-processo deve ser estável em relação à sua estrutura básica.
6. Gerenciabilidade de interface: Interfacear todos os processos por ele definido provendo, assim, um mecanismo de gerenciamento de inconsistências no momento em que estas são detectadas.
7. Integração: Integrar e suportar a tecnologia existente, as práticas de trabalho e os padrões de comunicação entre os envolvidos com o processo.
8. Suporte ao usuário: O meta-processo deve ser tolerante a falha e ser capaz de prover ajuda suficiente ao usuário, durante a execução de suas meta-atividades.
9. Benefício econômico: Possibilitar a melhoria da qualidade e produtividade do processo por ele criado e, conseqüentemente, do produto.
10. Implementabilidade: O engenheiro de processo, de posse do meta-processo, deve ser capaz de iniciar a implementação de um processo de produção de software na tecnologia corrente. O meta-processo deve estar aberto aos avanços tecnológicos.
11. Representatividade: A capacidade de interagir com os participantes do processo e alterar seu estado de representatividade, principalmente, na questão visual é um requisito pertinente a um meta-processo.

O trabalho de AVRILLIONIS et. al. (1999) não é o único a tratar da importância dos requisitos na definição de um meta-processo. REIS (2002) apresenta, em seus estudos, os requisitos para apoiar a reutilização de processos de produção de software. Para o autor, os requisitos devem ser divididos em 4 categorias:

1. Modelagem de processo visando à reutilização.
2. Recuperação e adaptação de processo.
3. Execução do processo.
4. Generalização e avaliação de processos encerrados.

1. **Modelagem de processos visando a reutilização.** Categoria que tem como objetivo obter modelos de processo genéricos que sejam úteis. Os modelos devem ser empacotados e disponibilizados em estruturas uniformes classificadas como *templates*. Esta categoria define que os *templates* devem ser encarados como modelos genéricos, utilizados como ponto de partida para a construção de um processo. A construção de tais *templates* é influenciada pelos seguintes requisitos:

- a. Definir os operadores utilizados na abstração e na modelagem de processos. REIS (2002) destaca a existência de dois operadores que podem ser utilizados na abstração e na modelagem de processos: composição e herança (operadores advindos do paradigma orientado a objetos). A composição é utilizada para compor estruturas de dados menores gerando, assim, modelos e *templates*. A herança é utilizada na definição e visualização de hierarquias entre os *templates*. Além dos operadores propostos por REIS (2002), PERRY (1996) argumenta que a abstração de processo utiliza técnicas de:

- i. Parametrização: Permite a definição abstrata de todos os elementos do processo, por exemplo: valores, tipo, atividades, tarefas e ferramentas.
 - ii. *Primitivation* (Primitivação): Técnica de abstração que requer a elaboração e a aplicação do processo antes de sua execução.
 - iii. Estratificação: Técnica de abstração que define processos em camadas, permitindo que determinadas partes sejam definidas somente com objetivos amplos, deixando sob a responsabilidade do usuário a implementação específica do processo. Nesta técnica, informações sobre ferramentas e métodos são excluídas da descrição do processo. A construção de um modelo com múltiplos níveis de detalhe, a partir das técnicas de composição, é uma constante nesta técnica.
- b. Separar os detalhes das múltiplas perspectivas: A representação, visual ou textual, de um processo deve atender as perspectivas de fluxo de trabalho, fluxo de dados, recursos e *skill* (perfil profissional ou habilidades) (DERNIAME et. al. (1999) e SOMMERVILLE (2003)). Existem várias linguagens de modelagem de processo, classificadas como híbridas, que permitem que a representação de um processo seja vista sob diferentes óticas. Estas linguagens podem facilitar o entendimento e potencializar a reusabilidade de um processo. Além da representação das perspectivas é necessário representar as camadas do processo, quando a técnica de estratificação é aplicada. REIS (2002) destaca que quanto maior o nível de detalhe de um processo, menor será o seu grau de reutilização e maior será o trabalho de representação. Em relação ao detalhamento do processo, PERRY (1997) mostra que, durante a modelagem, o projetista lida com uma estrutura multidimensional que mistura várias classes de informações, tais como: de projeto (domínio de conhecimento no qual o processo será aplicado); da organização (cargos e recursos utilizados para

percorrer as atividades do processo); do ambiente tecnológico (ferramentas utilizadas no desenvolvimento do software) e, por fim, do processo, que inclui informações sobre as atividades e estratégias gerenciais para atingir o sucesso no desenvolvimento do produto.

- c. Optar por modelar o processo com uma linguagem simples: Após definir o nível de detalhe de um processo, é necessário modelá-lo. Para isto, é necessário utilizar uma linguagem modelagem de processo que seja simples e inteligível. A utilização de uma linguagem que contenha elementos gráficos pode facilitar o entendimento do processo.

2. Recuperação e adaptação de processo (segundo requisito apontado por REIS (2002)): Geralmente, os modelos e os componentes de um processo estão armazenados em uma base de componentes (um repositório), traduzindo, assim, a necessidade de recuperar, selecionar e adaptar os componentes reutilizáveis de um processo. Segundo JØRGENSEN e CARLSEN (2001), existem várias formas de recuperação de um componente, por exemplo: recuperação de componentes procurando um texto que faça parte da sua composição estrutural; recuperação por meio da utilização de técnicas de busca em palavras chaves e; recuperação de um componente utilizando conceitos de navegabilidade nas estruturas hierárquicas de um processo. Já o conceito de adaptação traduz todas as modificações necessárias para reutilizar um processo recuperado. A adaptação envolve modificações sintáticas dentro do modelo de processo e define os envolvidos e os recursos necessários a partir do modelo original. Baseado nestas duas definições (recuperação e adaptação) apresentadas neste item, os requisitos utilizados para compor o conceito de recuperação e adaptação do processo são:

- a. Estruturar grandes bases de processos abstratos: O desenvolvimento ou a reutilização de um processo de software direciona, automaticamente, à existência de uma técnica de recuperação de ativos de processo (componentes, formulários e *templates*) a partir de critérios definidos pelo

projetista do processo. Este fato carrega consigo a necessidade de se estruturar as bases de processos para que a recuperação dos ativos seja rápida e consistente.

- b. Registrar a aplicabilidade do processo: Informações como pré-condições e recursos necessários para a instanciação de um processo devem estar registradas na base de processos abstratos. Além destas informações, as propriedades inerentes à aplicabilidade do processo devem estar estruturadas e, devidamente, armazenadas, facilitando assim, a reutilização do mesmo por parte do projetista de processo.
 - c. Extrair visões de modelos de processo: Extrair visões de processo se refere à exclusão de alguns detalhes, caracterizando assim, os processos gerados como modelos.
 - d. Adaptar o processo para contextos específicos: Requisito que contempla a adaptação de um componente a ser reutilizado ao domínio do projeto. Esta adaptação deve levar em conta as seguintes variáveis: contexto organizacional no qual o projeto de software está inserido e características do produto software.
3. **Executar o processo** (terceiro requisito apontado por REIS (2002)): Na execução do processo, o usuário pode obter apoio automatizado no desenvolvimento das atividades do processo. A execução de um processo deve ser alicerçada pela máquina de processo, estrutura esta que provê a comunicação dos indivíduos envolvidos com o mesmo. A máquina de processo deve possuir características adaptativas, pois a sua estrutura pode ser modificada, se o processo for adaptado a um ambiente não previsto na técnica de primitivação. Sendo assim, os itens abaixo apresentam os requisitos para apoiar a reutilização de processos durante a sua execução.

- a. Apoiar a evolução de processo: Modelos de processos são classificados como estruturas dinâmicas que podem sofrer alterações estruturais durante a sua adaptação. Para controlar tais alterações é indispensável à realização de um bom serviço de gerência de configuração e registro de decisões para eventos e modificações ocorridas no modelo de processo. Um mecanismo que possibilite ao usuário a navegação nas versões do modelo, a fim de entender a sua evolução histórica, é indispensável como método de apoio e importante dentro do contexto evolutivo do processo.
 - b. Compor dinamicamente os elementos do processo: Tendo em vista que um processo de produção de software é caracterizado pela composição dinâmica de seus elementos (atividades, tarefas, artefatos, envolvidos, ferramentas), se faz necessário, durante a sua execução, a estruturação de padrões para descrição de interfaces de ativos de software, apoiando às condições lógicas que direcionam a composição dos elementos quando estão em estado de execução.
4. **Generalização e avaliação de processos encerrados:** Quando um processo atinge o estado de encerrado, informações sobre o uso real das estratégias gerenciais adotadas pela organização durante a produção de software são armazenadas, por exemplo: uma base de métricas é estabelecida para o processo e para o projeto. Estas estratégias devem ser avaliadas e, se necessário, generalizadas para compor os modelos de processos armazenados no repositório de modelos (descrito anteriormente), com o objetivo de promover a evolução dos mesmos em uma próxima etapa. A seguir são apresentados os requisitos que apóiam a generalização e a avaliação dos processos encerrados.
- a. Generalizar processos encerrados: Possibilidade de descontextualizar o processo executado, removendo assim, os detalhes e aglutinando as diversas instâncias do mesmo em classes.

- b. Comparar a descontextualização com os modelos de processo: Sabendo que o conceito de generalização não é caracterizado como algo simplista, os usuários do processo devem comparar os modelos de processos existentes em seu repositório com o processo descontextualizado, para que a definição dos *templates* e as estratégias gerenciais inferidas respeitem um padrão e certo grau de granularidade.
- c. Conectar as instâncias aos processos abstratos generalizados: Quando um processo é descontextualizado, as instâncias geradas (entende-se por instâncias os *templates* e práticas gerenciais a serem refinadas ou adotadas) devem ser aglutinadas em classes. Estas classes devem ser conectadas aos modelos abstratos, provendo assim, o caráter evolutivo de tais modelos.

Apresentados os requisitos necessários para a definição de um meta-processo, segundo as propostas efetuadas por AVRILLIONIS et. al. (1999) e REIS (2002), a próxima seção deste trabalho irá apresentar os modelos de reuso de processos advindos da literatura. Tais modelos, assim como a teoria apresentada até ao momento, caracterizam o embrião conceutivo da proposta de tese apresentada no capítulo 5.

4.5 Modelos de Reuso de Processo Advindos da Literatura

4.5.1 – Modelo 3C de HOLLENBANCH e FRAKES

O modelo proposto por HOLLENBANCH e FRAKES (1996) pode ser considerado um clássico da literatura, fato este comprovado ao analisar os trabalhos de HENNINGER (1998), WILES (1999), NELSON (1999), GOMAA et. al. (2000) e FIORINI (2001). Em seu trabalho, HOLLENBANCH e FRAKES (1996) definem reuso de processo como um processo descritivo que pode ser utilizado na criação de outro processo, padronizando-o, sistematicamente, para adaptação e uso em um projeto específico. Na definição do modelo, os autores utilizaram as seguintes técnicas e

teorias: análise de domínio, princípios de reuso de software, processos, arquiteturas e mecanismos de modelagem de processo. HOLLENBANCH e FRAKES (1996) estruturam o modelo em 3 eixos: conceito, conteúdo e contexto (denominado modelo 3C):

- Eixo Conceitual: define uma especificação informal do processo, descrevendo as interfaces que o mesmo deve possuir quando colocado em execução. Uma descrição das características do cliente do processo também é desenvolvida neste eixo;
- Eixo de Conteúdo: inclui a descrição das tarefas do processo utilizando técnicas de representação gráfica e textual;
- Eixo de Contexto: descreve informações sobre o domínio no qual o processo será aplicado e sobre a organização no qual o processo está inserido. Custos e riscos são descritos neste eixo.

No modelo 3C os autores salientam que o objetivo de uma estrutura utilizada para a definição de processos (meta-processo), dentro de um contexto organizacional, deve possuir características adaptativas. Portanto, HOLLENBANCH e FRAKES (1996) apresentam, em seu trabalho, um conjunto de atividades utilizadas para a definição de um processo, e enfatizam a importância da atividade de adaptação no momento de sua instanciamento. As atividades propostas pelos autores são:

- Definir o processo como reutilizável (atividade ligada ao eixo conceitual e de conteúdo): A saída desta atividade é uma ou mais descrições de processos. Um norteamento adaptativo dos produtos de trabalho (artefatos e ativos do processo) deve contextualizar a execução desta atividade. A descrição do processo pode ser empacotada em manuais ou documentos descritivos;

- Desenvolver treinamento para o processo reutilizável (atividade de apoio): Os envolvidos com o contexto organizacional devem ser treinados no processo reutilizável;
- Adaptar o processo reutilizável em um projeto (atividade ligada ao eixo de contexto): O processo reutilizável é adaptado para reunir requisitos específicos no ambiente do projeto. Uma descrição do processo aplicado ao projeto específico caracteriza o resultado desta atividade;
- Adaptar o treinamento do processo reutilizável para o processo adaptado (atividade de apoio): Os envolvidos com a execução do processo devem ser treinados no processo adaptado;
- Executar o processo (atividade ligada ao eixo de contexto): O processo é colocado em prática, sendo que um acompanhamento relacionado às métricas da qualidade deve ser realizado durante a execução;
- Refinar o processo (atividade relacionada à melhoria): Baseado nas métricas coletadas e nos passos anteriores o processo é avaliado para verificar se o mesmo atingiu as metas iniciais propostas. Caso o processo não tenha atingido tais metas, uma análise é realizada, o processo é refinado e as atividades de treinamento e execução são desenvolvidas novamente.

Uma visão gráfica do relacionamento das atividades propostas no modelo 3C é apresentada por meio da Figura 4.13.

A composição estrutural das atividades apresentada na figura possui uma forte relação com o *framework* modular de reuso proposto por FRANCH e RIBÓ (2002) representado na Figura 4.12. Em seu trabalho, HOLLENBANCH e FRANKES (1996) mostram que a saída da atividade “definir o processo reutilizável” resulta na descrição de um ou mais processos, apologia à idéia de composição de processo proposta por FRANCH e RIBÓ (2002). Após a definição do processo,

HOLLENBANCH e FRAKES (1996) estabelecem a atividade de adaptação, apologia aos fluxos de adaptação propostos por FRANCH e RIBÓ (2002). Por fim, a idéia de melhoria, também, é perceptível na atividade de refinamento de processo proposta por HOLLENBANCH e FRAKES (1996).

Figura 4.13 – Relacionamento das atividades proposta no modelo 3C (adaptado de HOLLENBANCH e FRAKES (1996))

Além das atividades apresentadas na Figura 4.13 HOLLENBANCH e FRAKES (1996), formatam um conjunto de métodos de criação (ou descrição) de processos reutilizáveis. Tais métodos atuam dentro da atividade definir o processo reutilizável e são descritos a seguir:

- **Análise dos requisitos do processo:** Método responsável por definir as condições operacionais, na qual o processo irá executar, além de metas e requisitos que devem ser satisfeitos;

- **Análise de processo:** Método responsável por apoiar as definições dos processos genéricos reutilizáveis. As definições de tais processos ocorrem mediante o exame do projeto a ser desenvolvido. O projeto é examinado, e se existirem processos que se encaixem ao domínio procurado (tais processos devem possuir uma estrutura operacional semelhante ao processo que será aplicado ao projeto), estes são instanciados para, posteriormente, serem utilizados na proposta do novo processo, na qual este será aplicado ao projeto;
- **Projeto preliminar do processo:** Método responsável por definir os atributos do processo: clientes, entradas, saídas e indicadores da qualidade;
- **Projeto detalhado do processo:** Método responsável por definir as tarefas, métodos, procedimentos, papéis e mecanismo de controle pertinentes às atividades do processo;
- **Codificação:** Após o processo ser definido ele deve ser executado;
- **Teste e integração:** Método que garante que o processo definido está aderente às metas, anteriormente, definidas. A atividade de teste e integração é interpretada como uma simulação do processo.

4.5.2 – Modelo Gertude de SUCCI, BENEDICENTI, PREDONZANI e VERNAZZA

O modelo Gertude foi proposto por SUCCI et. al. (1997) e está alicerçado na teoria de reuso de processo de software. Em seus estudos, os autores definem que o reuso de processo de software requer a configuração de um repositório, na qual os usuários podem recuperar a descrição de um determinado processo de projeto. Para eles, a idéia de reuso do processo requer, do ponto de vista do usuário, a facilidade de compreensão de objetos de um determinado domínio do conhecimento, fato este que isenta a definição de um modelo com um alto grau de formalismo. Porém, do ponto de vista de um repositório, o processo reutilizável deve ser especificado, formalmente, o que remete à elaboração de técnicas para seu armazenamento e

recuperação. O modelo Gertude atende os dois pontos de vista e organiza os objetos de processo em quatro categorias, chamadas de entidades: pessoas, regras, atividades e infra-estrutura.

As atividades se caracterizam como a parte central do modelo, sendo que a agregação delas constituem o processo. As pessoas executam as atividades de acordo com um determinado conjunto de regras. A infra-estrutura é caracterizada pelos equipamentos e as instalações necessárias para a execução das atividades por parte das pessoas.

As atividades, por sua vez, são classificadas em 3 categorias: 1 – atividades de interface; 2 – atividades de controle e 3 – atividades de execução. As atividades de interface envolvem, diretamente, o relacionamento dos clientes externos com o contexto organizacional. As de controle têm como objetivo supervisionar e coordenar o trabalho do processo em questão, sendo assim, as atividades relacionadas a este cunho estão ligadas ao âmbito organizacional, como demonstrado na Figura 2.3. As atividades de execução têm como objetivo executar as tarefas específicas do processo transformador.

Além da classificação das atividades em categorias, o modelo prevê o desenvolvimento de uma base de dados histórica, cujo objetivo é armazenar os dados sobre a execução de um processo instanciado. A principal finalidade desta base de dados é fornecer subsídios para avaliação do desempenho e, se necessário, o processo deve ser ajustado. A calibração do processo está, intimamente, ligada à vertente da qualidade inferida a partir dos estudos desenvolvidos por REIS (2002).

Uma das preocupações do Gertude é a forma de acesso ao processo por parte da entidade pessoas. Algumas pessoas podem ter acesso a todo o processo, por exemplo: o engenheiro de processo. Porém outras necessitam acessar, somente, uma parte do processo que está relacionada com as suas

responsabilidades, por exemplo: o programador dever possuir acesso somente as atividades do processo que tratam os aspectos da codificação.

Já em relação à descrição do processo, o Gertude prevê a utilização de duas linguagens:

- uma interna, cujo objetivo é modelar o processo, possibilitando, assim, a recuperação de uma determinada descrição dentro do repositório. A linguagem interna possui características técnicas, completas e formais, o que condiz com as colocações sobre a teoria de processo reutilizável, apresentada no início desta seção;
- e outra externa, notação informal utilizada para que as pessoas consigam ter uma visão simples e sintética do processo.

Além de descrever as principais características do modelo Gertude, SUCCI et. al. (1997) destacam que a idéia de reutilização de processo passa por 4 etapas: descrição, classificação, recuperação e adaptação. Tais etapas também são configuradas nos trabalhos de FRANCH e RIBÓ (2002), REIS (2002) e FIORINI (2001). Em seu trabalho, SUCCI et. al. (1997) focam somente os aspectos descritivos de um processo. Segundo eles, os aspectos classificatórios e de recuperação são necessários somente quando vários processos estão armazenados em um repositório. Já o aspecto adaptativo não está formalizado em seu trabalho

Na proposta do Gertude, a descrição de um processo de software passa por 4 fases distintas:

- a. Modelagem dinâmica: Neste tipo de modelagem é possível perceber o processo em execução. A técnica de desenvolvimento de cenários, posicionando atores e objetos dentro de um processo real, é utilizada neste tipo de modelagem.

- b. Modelagem estática: Neste tipo de modelagem é possível perceber a representação hierárquica dos objetos que compõem o processo.
- c. Aquisição de dados: Na execução do processo, sobre um projeto, os dados são obtidos e coletados.
- d. Análise de dados: Os dados são planilhados, estatisticamente, e analisados.

A seguir, um exemplo⁴ apresentando a 4 fases distintas é configurado neste trabalho.

Fase a - Modelagem Dinâmica

Nesta subseção será apresentada a modelagem dinâmica de um processo de software. O domínio utilizado é uma empresa de projeto de software. O modelo de processo utilizado é o cascata.

Segundo SUCCI et. al. (1997), para realização da modelagem dinâmica do processo, primeiro é necessário descrever os seus casos de uso. Neste exemplo, foi verificado que existe 1 caso de uso descrevendo de forma global o processo de software. O Quadro 4.1, apresenta a descrição textual do caso de uso.

Quadro 4.1 – Caso de uso descrevendo parte de um processo de software. Adaptado de SUCCI et. al. (1997).

Quando Joaquim (o cliente) pede por um projeto, Diana dispara a coleta de seus requisitos. Diana informa Manuel sobre a existência de um novo projeto. Manuel utiliza a estratégia de alocação de recursos para escolher qual projetista irá trabalhar no projeto. Manuel possui 3 projetistas disponíveis (João , José e Maria). Ele escolhe José e Maria para trabalhar neste projeto. João é informado que irá trabalhar na validação do projeto também. Se o projeto não atender os requisitos, o projetista, João , terá que corrigi-lo. Quando o projeto for finalizado, Maria informa Diana , e esta entregará o produto para o cliente. Diana informa Manuel sobre a entrega. Manuel libera os projetistas.
--

⁴ O exemplo ilustrado nesta seção foi, integralmente, extraído do trabalho de SUCCI et. al. (1997). Somente tais autores, daqueles analisados, contemplam um exemplo ilustrativo.

De posse da descrição apresentada no Quadro 4.1, é possível identificar, facilmente, os envolvidos com o processo: Joaquim, Diana, Manuel, João, José e Maria. Além da descrição textual o caso de uso, também, é representado, graficamente (vide Figura 4.14), por um diagrama de seqüência (neste caso a Linguagem de Modelagem Unificada (UML) se torna a *Process Modeling Language* (PML)).

Analisando o Quadro 4.1 e a Figura 4.14 é possível estabelecer os papéis de cada participante no processo:

- Diana é analista: Atua na coleta de requisitos do software a ser construído;
- Manuel é gerente de projeto: responsável por alocar recursos para o projeto e controlar sua execução;
- João é projetista e atua na validação de projeto: Responsável por validar, com base nos requisitos, as atividades realizadas pelos projetistas;
- José e Maria são projetistas: Desenvolvem o projeto com base nos requisitos.

Com base nos papéis, as atividades desenvolvidas no processo de software em questão são:

- coleta e análise de requisitos (interação com o cliente) categorizada como atividade de interface;
- projeto de software categorizada como atividade interna;
- validação do projeto categorizada como atividade de controle e;
- gerenciamento e projeto categorizada como atividade de controle.

Figura 4.14 – Diagrama de seqüência ilustrando a execução de um processo de software – representação gráfica da descrição apresentada na Quadro 4.1. Adaptado de SUCCI et. al. (1997).

Fase b - Modelagem Estática

Nesta fase será apresentada a estrutura hierárquica dos objetos (papéis, atividades) inferidos na modelagem dinâmica. A técnica de modelagem utilizada nesta seção é baseada em organogramas, conforme vista na Figura 4.15.

a) Estrutura hierárquica do processo

b) Estrutura hierárquica de papéis

c) Relacionamento entre atividades e papéis

Figura 4.15 – Modelagem estática de um processo. Adaptado de Succì et. al. (1997).

Ao analisar a Figura 4.15(a) é possível verificar que o processo de produção de software, hierarquicamente, é definido da seguinte forma: o nó processo é composto de coleta e análise de dados, projeto de software, validação de projeto e gerência de projeto. A estrutura hierárquica dos papéis é verificada na Figura 4.15(b). Nesta figura um colaborador pode ser classificado como gerente de projeto (responsável por parte do processo organizacional da empresa), engenheiro de software, analista de sistemas ou projetista (todos estes imersos na visão processual de uma determinada organização). Já a Figura 4.15(c) estabelece a relação entre os papéis e as atividades do processo, destaque para as atividades de projeto de software e validação de projetos que requerem a utilização de projetistas e analistas em ambas.

Na modelagem estática não está presente o conceito pessoas. Este tipo de objeto é configurado de acordo com a disponibilidade dos recursos humanos no momento da configuração do processo.

Fases c e d– Aquisição dos Dados e Análise Estatística

Conforme citado anteriormente, no momento da execução do processo os dados sobre produtividade são coletados (vide seção 4.5.2). Cada pessoa deve informar, por exemplo, o tempo que levou para executar uma determinada atividade. Em seu trabalho SUCCI et. al. (1997) destacam que os dados devem ser coletados de forma simples e consistente. Os autores propõem a utilização de uma ferramenta de controle de produção para a realização da coleta, a qual esta deve fornecer, periodicamente, os dados consolidados, conforme verificado na Tabela 4.3.

Colaborador: Maria								
Departamento: Projeto de software				Semana: 17/3 a 23/3 (horas trabalhadas)				
Atividade	Seg	Ter	Qua	Qui	Sex	Sab	Dom	total
Projeto	7	7	3	5	6	0	0	28
Validação	1	0	5	2	1	0	0	9
Outros (avaliação de novas ferramentas case, por exemplo)	0	1	0	1	0	0	0	2
Ocioso	0	0	0	0	1	0	0	1
Total	8	8	8	8	8	0	0	40

Tabela 4.3 – Dados consolidados de produtividade de uma pessoa. Adaptado de Succi et. al. (1997).

De posse dos dados, é possível desenvolver uma análise estatística, por pessoa, por departamento ou por projeto. Um exemplo desta análise pode ser verificado na Tabela 4.4.

A modelagem do processo de produção de software desenvolvida por SUCCI et. al. (1997), neste exemplo, pode ser armazenada em um repositório, posteriormente, recuperada e adaptada para um novo projeto, fato este que consolida a idéia de reutilização do processo.

Colaborador	Maria
Departamento	Projeto
Atividade	% tempo
Projeto	70
Validação	22,5
Avaliação de novas ferramentas case	5
Ocioso	2,5
Total	100

Tabela 4.4 - Análises estatísticas sobre a produtividade de uma pessoa. Adaptado de Succi et. al. (1997).

4.5.3 – Modelo de BECKER, HAMANN E VERLAGE

Em seu trabalho, BECKER et. al. (1997) ressaltam que as atividades e tarefas relacionadas ao desenvolvimento de software são, geralmente, caracterizadas como complexas. Em virtude deste fato, várias pesquisas têm como objetivo desenvolver e manter um processo de software que possa ser replicado e reutilizado em vários projetos. Segundo os autores, a idéia de desenvolvimento, manutenção, replicação e reutilização de processos, quando unidas e aplicadas ao domínio da engenharia de software, são definidas, unicamente, como modelo de processo. Eles relatam que um modelo de processo tem que criar uma representação de processos do “mundo real”. Esta representação, por sua vez, deve focar os seguintes objetivos:

- Medir, quantitativamente, produtos gerados pelo processo e recursos consumidos quando o mesmo é percorrido. Os dados gerados devem ser empregados na melhoria do processo;
- Habilitar a comunicação fora dos limites do projeto: Quando os modelos são instanciados em vários projetos, dados sobre produtos criados e recursos consumidos são gerados. Estas informações devem habilitar a comunicação entre os envolvidos nos projetos, com o objetivo de compartilhar experiências;
- Gerenciar mudanças: Os modelos de processos servem como base para analisar a implementação de mudanças no processo. A análise pode ser realizada de forma estática (avaliação e discussão dos dados gerados após a execução do processo) ou dinâmica (avaliação com o processo em execução e, se necessário, alterações no processo em andamento são implementadas).

Os autores deixam claro que um processo pode ser visto sobre duas óticas, descritiva e prescritiva. A visão descritiva determina como um processo é executado em um determinado ambiente ou domínio do conhecimento (em um projeto). Já a visão prescritiva apresenta como um processo deve ser executado não levando em consideração as variáveis que compõem o ambiente. Na visão descritiva, é

necessário modelar a aderência do processo ao projeto, e na visão prescritiva o foco está nas características de abstração, possibilitando assim que o processo possa ser prescrito e, posteriormente, descrito para um projeto.

BECKER et. al. (1997) focam somente a visão descritiva em seu trabalho, salientando que, para se modelar um processo, é necessário envolver dois tipos de atores (ou papéis): agentes de processo e engenheiros de processo. Os agentes de processo são responsáveis pela execução do processo, neste contexto se encaixam os programadores, os analistas de negócio e de sistema, engenheiros de software e demais atores que se relacionem, diretamente, com o processo. Os engenheiros de processo têm, como responsabilidade, o desenvolvimento e a melhoria do modelo de processo, nas empresas produtoras de software este papel é desempenhado pelo *Software Engineering Process Group* (SEPG) ou Grupo de Processos de Engenharia de Software.

O modelo de reuso proposto pelos autores está dividido em duas fases: configuração e execução, cada uma contendo 4 passos para o desenvolvimento:

- **Fase 1 – Configuração:** Nesta fase a modelagem de processo é preparada, o escopo contextual do processo é definido e as ferramentas são selecionadas.
 - Passo 1 – Definir escopo e objetivos: A modelagem em uma empresa inicia com o mapeamento do escopo e dos objetivos do modelo de processo. É importante conhecer o que o modelo de processo vai usar, as métricas são entendidas e os envolvidos com o processo são mapeados;
 - Passo 2 – Selecionar ou desenvolver um esquema de modelagem: Este esquema deve identificar e estruturar a informação capturada por meio do modelo de processo. Os conceitos para descrever o modelo de processo dependem dos objetivos definidos no passo 1;

- Passo 3 – Selecionar uma linguagem de modelagem de processo (PML). A necessidade de descrever o processo, formalmente, é uma constante na teoria de reutilização e meta-processo. A descrição formal de um modelo de processo passa pela definição de uma PML. A escolha de uma PML deve envolver os seguintes critérios: domínio de aplicação, no qual o processo será executado, notação gráfica simples, concisa e consistente e possibilidade de desenvolver formalizações textuais;
- Passo 4 – Selecionar e adaptar ferramentas: Dependendo da PML selecionada no passo anterior, ferramentas de suporte ao formalismo devem ser selecionadas ou adaptadas. O uso de ferramentas é crucial para que a modelagem do processo seja eficiente em relação à produtividade e inteligibilidade. Análise de ferramentas que suportam notações gráficas constituem uma boa pedida para o desenvolvimento deste passo. Alguns critérios para a escolha de ferramentas são contemplados pelos autores deste modelo, entre eles é possível destacar:
 - Simbologia: a ferramenta deve conter todos os símbolos definidos no passo 3;
 - Portabilidade: possibilidade de exportar o processo modelado para várias linguagens, principalmente, o *Hipertext Markup Language* (HTML) (Linguagem de Marcação de Hipertexto);
 - Consistência: a ferramenta deve impor requisitos de consistência no relacionamento dos símbolos. Por exemplo: não é possível relacionar um determinado ator com um artefato sem a presença de uma atividade ou tarefa que interligue ambos.

- **Fase 2 – Execução:** Nesta fase o processo modelado é executado em um projeto.
 - Passo 1 – Levantar os requisitos do processo: O objetivo deste passo é adquirir as informações (sobre agentes (atores), atividades e produtos, bem como qualquer relacionamento entre eles) necessárias para a descrição do processo de software (a instanciação do modelo). Para executar este passo, é necessário utilizar algumas técnicas, tais como: entrevistas com pessoas, observação do ambiente de produção de software, análise de protocolos de comunicação entre os agentes, análise de documentos e produtos. Na atividade de levantamento de requisitos do processo não se deve julgar nem o processo, nem os atores responsáveis pela sua execução. Quando uma pessoa desconfia que a sua forma de trabalho está sendo julgada, ela não irá compartilhar todas informações necessárias para a definição de requisitos do processo. A descrição do processo de software obtém informações de como o processo está sendo executado e não como deveria ser executado;
 - Passo 2 – Criar modelo. Identificados os requisitos do processo é necessário criar o modelo. Neste passo as atividades, os fluxos de informações e de controle, e as responsabilidades anexadas às atividades devem ser modeladas. Após a execução deste passo o modelo é validado junto aos participantes ativos do processo. É importante que vários ciclos de criação e validação sejam desenvolvidos neste passo, até que um modelo que atenda os requisitos de todos os agentes seja produzido;
 - Passo 3 – Análise do modelo de processo: O objetivo deste passo é detectar inconsistência no modelo. A análise do processo está centrada em duas propostas:

- Verificar a melhor forma de executar o processo;
- Identificar pontos de melhoria.

É necessário assegurar que os aspectos sintáticos e semânticos do processo estejam consistentes. Riscos, fatores críticos de sucesso e *deadlock*⁵ devem ser mapeados neste passo. Os critérios de análise do processo estão divididos em duas categorias:

- Análise estática: verifica a “completude” (todas as informações necessárias para obter os objetivos foram incluídas no modelo), “corretude” (o modelo não contém informações contraditórias) e consistência estrutural do processo (utilização de regras na verificação, por exemplo: se uma atividade é refinada então todos os produtos usados pela atividade são utilizados);
- Análise dinâmica: verifica o processo durante sua execução, identifica possíveis situações de *deadlock* e transições de estados não ambíguas.
- Passo 4 – Análise do processo: Este passo tem como objetivo analisar os aspectos quantitativos e qualitativos. Os aspectos quantitativos focam a correlação entre diferentes atributos do processo, baseadas em medidas que foram geradas durante a sua execução. Já os aspectos qualitativos focam questões relacionadas à concentração de responsabilidades em um determinado agente, excesso de *feedback* e alto grau de dependências entre as atividades.

⁵ Segundo MACHADO e MAIA (2002), *deadlock* é a situação em que um processo, atividade ou tarefa aguarda por um recurso (artefato, informação) que nunca estará disponível. Esta situação pode ocorrer nos processos que compartilham artefatos e informações.

4.5.4 – Modelo de HENNINGER

O modelo apresentado por HENNINGER (1998) defende que a idéia de reutilização ou definição de um processo de produção de software passa pela descrição e armazenamento de estruturas de dados que possibilitem a instanciación de um processo. O autor salienta, ainda, que o conceito de instanciación está ligado à aderência que o processo deve ter a projetos de softwares específicos. A configuração arquitetônica do modelo está dividida em 5 compartimentos (vide Figura 4.16):

- Repositório de processo: Neste compartimento são armazenados os modelos de processo que serão utilizados em um determinado projeto. Segundo o autor, o modelo de processo é composto por um conjunto de atividades, artefatos e ferramentas. Além destes modelos, as informações (características) dos projetos, também, são armazenadas. Dados como tempo que o projeto levou para ser concluído, o processo utilizado para o seu desenvolvimento, recursos consumidos e as lições aprendidas são armazenados nesta base de dados. De posse das informações dos processos, de seus modelos e de um estímulo externo, que representa necessidades de desenvolvimento, um processo é selecionado pelo componente de seleção;
- Processo adaptado: Compartimento resultante do componente de seleção de processo. Deve ser aderente ao projeto que será desenvolvido;
- Projeto de software: Relaciona as atividades, tarefas, artefatos, recursos e ferramentas para o desenvolvimento de um projeto. O projeto deve ser planejado, executado e controlado;
- Revisão do processo: Compartimento composto por um grupo de atores cujo objetivo é verificar se o processo definido (ou reutilizado) antighu as metas propostas no projeto. Na maioria das empresas este grupo é caracterizado como grupo de engenharia e processo de software.

- Processo modificado: Se no compartimento anterior for detectada a necessidade de alteração do processo, segundo as concepções definidas no projeto, o processo é modificado e armazenado na base de dados de modelo de processo.

Figura 4.16 – Estrutura arquitetônica do modelo proposto por Henninger. Adaptado do HENNINGER (1998).

4.5.5 – O Modelo Arquitetônico de Reutilização de Processos de FIORINI

FIORINI (2001) apresenta, em seu trabalho, uma arquitetura para reuso de processos cujo objetivo é organizar e viabilizar estruturas relacionadas ao processo de software para sua reutilização. O modelo proposto por FIORINI (2001) (vide Figura 4.17) é composto de:

Figura 4.17 – Modelo arquitetônico para reutilização de processos (retirado de FIORINI (2001), página 46).

1. um **framework de processos**, definido como um processo genérico que tem como meta, gerar novos processos. Segundo FIORINI (2001 p. 55):

“O framework é uma generalização de um determinado conjunto de processos de um determinado domínio. Ele possui uma parte fixa e uma variável. A fixa é composta de atividades comuns à maioria dos processos que podem ser instanciadas pelo framework. A parte variável (hot-spot) é composta de atividades e elementos de processos que definem características ou caminhos específicos de uma instância do framework. Os hot-spots são preenchidos pelo engenheiro de processos, selecionando componentes existentes ou redefinindo/definindo a descrição dos elementos do processo”.

2. um **guia de reutilização** que tem, como meta, ajudar o engenheiro de processo na reutilização e instanciação do *framework*. A autora salienta que existem dois tipos de guias, o do *framework* que fornece uma visão ampla e genérica das atividades; e o das atividades, cujo objetivo é fornecer uma visão detalhada e

específica de cada atividade do processo. Uma visão conceitual do guia de reutilização proposto por Fiorini pode ser verificada na Tabela 4.5.

Nome da Atividade ou do Processo:			
<u>Conceito:</u> () – Framework		() Atividade	
		() – Ponto Comum	
		() – Ponto de Flexibilização	
<u>Conteúdo:</u>			
Atividade	Características	Componentes	Processos Standard ⁶ / Patterns
<<nome atv. 1>>	E	Componente X	
<<nome atv. 2>>	OP	Componente Y	
<<nome atv. N>>	RE		
<u>Contexto:</u>			
Situações do contexto Usar			

Tabela 4.5 – Guia de Reutilização (retirado de FIORINI (2001), página 57)

Baseado na tabela acima, é perceptível que o guia de reutilização proposto pela autora é baseado na proposta do modelo 3C desenvolvida por HOLLENBANCH e FRAKES (1996) – CONCEITO, CONTEÚDO e CONTEXTO.

O eixo conceitual define se o guia é classificado como *framework* ou de atividades. Se o guia for classificado como atividades, os pontos de flexibilização também são identificados.

No eixo conteúdo é identificado, para cada atividade: os componentes e características. Os componentes podem ser classificados como ferramentas, *templates* ou ativos de processos que podem ser utilizados para a execução de uma atividade. Já as características, são identificadas a partir das informações da análise do domínio da aplicação do processo, as quais classificam as atividades como:

- Comuns [CO]: Atividades que aparecem em mais de 75% dos processos analisados (processos estes armazenados no módulo ou compartimento de tipos de processos);
- Opcional [OP]: Atividades com baixo índice de ocorrência nos processos;

⁶ Processo *standard* pode ser utilizado para a definição de novos processos de produção de software.

- Essencial [E]: Atividade que possui muita importância dentro processo e classificada como fundamental. A importância é medida pelo índice de frequência (>75%) com que ele está presente nos processos analisados;
- Recomendável [REC]: Quando a atividade não possui frequência muito alta (entre 74% e 50%), mas causa influência no domínio;
- Analisada [AN]: Atividade com frequência entre 49% a 25% em relação ao domínio do processo deve ser analisada de forma a verificar se a mesma pode contribuir para o processo que está sendo definido;
- Específica [E]: Quando a atividade está presente somente em poucos processos do domínio. Uma atividade específica representa uma particularidade de um processo.

Segundo FIORINI (2001), no eixo contextual do guia do *framework*, é possível encontrar as descrições de “situações de contexto”. A situação de contexto define o estado ou ambiente no qual o projeto está incluso ou que os atores (responsáveis pelo projeto) estão passando.

3. **Tipos de processos:** banco de dados de processos *patterns*, usual e *standard*;
4. **Linguagens de modelagens de processos (PMLs):** Estas linguagens têm como objetivo oferecer recursos para descrever e manipular as atividades do processo genérico (*framework*) e dos processos *patterns*, usual e *standard*.

4.6 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo

Este capítulo apresentou um levantamento bibliográfico sobre processo de software, modelos de processos, meta-processo e reuso de processo.

Ao analisar o referencial bibliográfico que norteia os conceitos de meta-processo e reuso de processos, o autor deste trabalho constata que: Ambas teorias não são excludentes, mas sim complementares. Tanto a teoria de meta-processo quanto a de reuso de processo partem de estruturas pré-concebidas para a criação ou organização de um processo de produção de software. Este fato motivou a inserção do embasamento bibliográfico em questão, a fim de alicerçar a proposta de criação e a organização de um processo de produção de software em um contexto fabril. A proposta preliminar deste modelo será apresentada no capítulo 5.

Capítulo 5 – Proposta Preliminar do Modelo para a Criação e a Organização de Processos de Produção em um Contexto de Fábrica de Software

Este capítulo tem como objetivo apresentar a proposta preliminar¹ do modelo para a criação, organização, instanciação e adaptação de processos de produção de software para o contexto fabril.

A proposta embrionária do modelo está baseada na definição de processo transformador e nas teorias de meta-processo e de reuso de processo de software, apresentadas no capítulo 4. Em tal capítulo é perceptível que:

1. Ambas as teorias (meta-processo e reuso de processo) são complementares.
2. Um processo de produção é composto por:
 - o entradas, classificadas como matéria prima, recursos e mão de obra.
 - o e a saída, definida como produto.

De posse da informação 1, é possível afirmar que o modelo foi concebido sobre 3 visões: visão de reuso; visão de definição e; visão compartilhada (visão esta que trata de questões de execução, avaliação e armazenamento do processo, atividades necessárias quando se cria ou se reutiliza um processo de software). Salienta-se que a proposta apresentada neste trabalho une a teoria de reuso de processo apresentada por SUCCI et. al. (1997), BECKER et. al. (1997), HENNINGER (1998), JØRGENSEN e CARLSEN (2001), FIORINI (2001), FRANCH

¹ A proposta é considerada preliminar, pois a mesma incorpora apenas os relatos bibliográficos apresenados nos capítulos 2, 3 e 4. A aderência da proposta aos processos de produção das empresas caracterizadas como fábrica irá produzir alteração em sua estrutura.

e RIBÓ (2002) e REIS (2002), à teoria de meta-processo delineada por DERNIAME et. al. (1999) e AVRILLIONIS et. al. (1999) (vide Figura 5.1).

Visão 1 – VISÃO DE REUSO	Visão 2 – VISÃO DE DEFINIÇÃO
Visão 3 – VISÃO COMPARTILHADA	

Figura 5.1 - Visões utilizadas na concepção do modelo

O produto gerado pelo modelo é um processo de software com características fabris, fato este explicitado, visualmente, por meio da Figura 5.2. Em tal figura, salienta-se a presença do ator engenheiro de processo², pessoa que percorre, diretamente, a estrutura proposta no modelo, sendo responsável por todas as definições incorporadas ao processo de produção de software. Ao analisar a figura em questão, é possível dizer que o modelo possui:

- entradas, caracterizadas como requisitos de um processo de software;
- processo transformador, responsável pela criação, organização, instanciação e adaptação do processo de software e;
- saída: processo de produção de software instanciado, que será percorrido quando um projeto for executado.

Figura 5.2 – Visão embrionária do modelo (Adaptado de FABRI et. al. (2007b))

² Nas empresas de desenvolvimento de software, o papel do engenheiro de processo é representado pelo *Software Engineering Process Group* (SEPG) ou Grupo de Processo e Engenharia de Software.

Na definição dos requisitos³ o modelo deve estabelecer o domínio de atuação de uma fábrica de software. PRESSMAN (2002) apresenta em sua obra, um exemplo que retrata tal questão. Para autor um processo para desenvolver software para o controle de uma aeronave (domínio de atuação A) é, totalmente, diferente de um processo para desenvolver um software para gestão de uma universidade (domínio de atuação B). Um outro exemplo de uma fábrica de software orientada a domínios, focada no desenvolvimento de aplicações distribuídas para dispositivos pervasivos (BURKHARDT et. al. (2002)), é apresentada por FABRI et. al. (2005). Para o desenvolvimento de sua proposta, os autores partem da idéia de linha de produto de software⁴ unindo-a a segmentação pervasiva citada. PERRY (1997) também salienta a importância de se criar um processo de produção de software focado em um domínio de conhecimento pré-estabelecido. Em seu trabalho o autor mostra que durante a definição de um processo, o engenheiro lida com uma estrutura multidimensional que mistura várias classes de informações, tais como a de projeto (domínio de conhecimento que o processo será aplicado).

Arelada à definição do domínio de conhecimento, está à configuração tecnológica da fábrica, por exemplo: não seria interessante utilizar como tecnologia, a linguagem Clipper 5 (PRATES (1996)), no desenvolvimento de dispositivos pervasivos. Na configuração do eixo tecnológico, também, se encontram imersas as ferramentas utilizadas dentro do ambiente de produção, por exemplo: ferramenta para produção de software; gestão de projetos; testes; controle de produção e gestão de configuração.

Um outro requisito importante a ser mapeado no contexto do modelo é o ciclo de produção no qual a fábrica de software está inserida. Na proposta deste ciclo, este trabalho se baseia nas considerações efetuadas por: FABRI et. al. (2004, 2004a e 2004b) e TRINDADE (2006); e na extensão do modelo classificatório proposto por FERNANDES e TEIXEIRA (2004), inferido no capítulo 3 (vide Figura 3.7). De posse de

³ A idéia de levantamento de requisitos de um processo, também, está presente no modelo 3C de HOLLENBANCH e FRAKES (1996) – vide capítulo 4.

⁴ CLEMENTS E NORTHROP (2002), definem que uma linha de produto de software é um conjunto de sistemas que usam software, intensivamente, compartilhando as características comuns, satisfazendo as necessidades de um segmento particular do mercado, e são desenvolvidos a partir de um conjunto comum de ativos.

tais informações, é possível definir que uma fábrica de software pode se configurar nos seguintes ciclos: longo, médio e curto. A definição para tais ciclos foi efetuada na seção 2.1.2.

Atrelados à definição do ciclo de produção, encontram-se os aspectos da modelagem do processo. Neste tipo de modelagem, as relações dinâmicas entre as atividades são definidas. Sendo assim, é necessário verificar se a fábrica irá operar no modelo cascata, incremental, espiral ou prototipagem entre outros. FABRI et. al. (2007) salientam que a definição do modelo de processo influenciará no comportamento do processo fabril, sabendo que o mesmo define como as atividades do processo estarão se relacionando sob a ótica temporal.

Por fim, o modelo da qualidade que irá compor o processo de produção de software deve ser definido no levantamento de requisitos. FIORINI (2001), define tais modelos como processos *standard*. Em seu trabalho a autora relata que um processo *standard* pode ser utilizado para a definição de novos processos de produção de software. Neste trabalho os modelos da qualidade são interpretados de uma outra maneira; segundo FABRI et. al. (2007), a proposta efetuada neste capítulo absorve tais modelos em sua estrutura interna e eles são utilizados para revestir as atividades e tarefas definidas nos ciclos de produção: longo, médio e curto. Os autores deixam claro que a criação e a organização do processo são de responsabilidade da teoria de meta-processo ou de reuso de processo, sendo que os modelos da qualidade são definidos como guias e estruturas de apoio. Visão esta compartilhada pelo modelo CMMI, no qual é caracterizado como um guia utilizado na melhoria do processo de produção de software, sendo que, aspectos de gerenciamento e manutenção de produtos e serviços não são contemplados. O CMMI não pode ser caracterizado como um processo devido à falta de algumas características como: definição de domínios de aplicações, modelagem tecnológica do ambiente; modelos configuração de ativos (ou *templates*) e; estrutura organizacional (CMMI-SE/SW (2002) e CMMI-SE/SW (2002a)).

Levantados os requisitos de processo fabril para o desenvolvimento de software, o modelo entra em fase de execução, com isso as seguintes atividades devem ser percorridas:

1. Modelagem.
2. Instanciação.
3. Simulação.
4. Execução.
5. Avaliação.
6. Armazenamento.

5.1 Atividade de Modelagem⁵

Nesta atividade é necessário modelar o processo abstrato utilizando uma linguagem de modelagem de processo. Segundo REIS (2002), a PML deve possuir um conjunto de operadores que suporte a modelagem de processos abstratos. Em seus estudos PERRY (1996) e REIS (2002), apresentam dois tipos de operadores relacionados a abstração de processos: composição e herança, operadores estes incorporados nas técnicas de modelagem utilizadas neste trabalho. Além dos operadores, as técnicas de parametrização, *primitivation* e estratificação, também, devem ser utilizadas nesta atividade. Para a modelagem de um processo, este trabalho propõe, que seja utilizada uma adaptação do atigrama, definido pela normalização IDEF-0⁶, representada na Figura 5.3. O fator motivador de tal escolha pode ser verificado no trabalho de FABRIL et. al. (2007a)⁷.

⁵ Atividade presente nos modelos 3C de HOLLENBANCH e FRAKES; Gertude de SUCCI, BENEDICENTI, PREDONZANI e VERNAZZA; (BHV) de BECKER, HAMANN E VERLAGE; e no (F) Modelo Arquitetônico de Reutilização de Processos de FIORINI – vide capítulo 4.

⁶ A normalização IDEF-0 é, amplamente, utilizada na modelagem de sistemas e processos de negócios. O IDEF-0 foi desenvolvido a partir de uma necessidade da força aérea americana, que trabalhava com diversas

Ao analisar a Figura 5.3, é possível verificar a presença:

- de um quadro, representando a atividade ou tarefa de um processo;
- dos artefatos de entrada e de saída;
- das regras que norteiam a execução da atividade ou tarefa;
- das ferramentas utilizadas, cujo objetivo é auxiliar na execução do processo;
- e das habilidades⁸ necessárias para que a atividade seja executada com um alto padrão de qualidade.

Figura 5.3 – ICOM representando o padrão para modelagem de processo

indústrias aeroespaciais. Em 1972, foi desenvolvido o SADT (*Structured Analysis and Design Technique*) por Douglas T. Ross, da *SoftTech*, que foi utilizado no projeto AFCAM (*Air Force Computer Aided Manufacturing*). Em seguida foram desenvolvidos o ICAM I (*Integrated Computer- Aided Manufacturing*) e o ICAM 2. O ICAM 2 foi documentado e renomeado como IDEF-0. O IDEF-0 processa uma coleção de atividades utilizando-se de ICOMs (*Input Control Output Mechanism*). O ICOM ou atigrama não inclui apenas dados e informações, mas também tudo que pode ser descrito como sendo um processo (esquema, estimativa, regulamentos e produtos). Para maiores informações sobre o IDEF-0, consulte a referência IDEF0 (1993).

⁷ Segundo os autores citados o atigrama, utilizado como notação na atividade de modelagem, proporciona a visualização do processo de software sob as vertentes de fluxo de trabalho, fluxo de dados, das habilidades e ferramentas.

⁸ O fluxo habilidades não está presente na versão original do IDEF-0, fato este que levou o autor deste trabalho a utilizar o termo “adaptação do diagrama IDEF-0”.

Ao analisar a parte interna da ICOM (*Input Control Output Mechanism*), é possível verificar que as atividades podem ser ligadas uma as outras, por meio das operações de composição e de herança, operadores estes, apresentados nos estudos de PERRY (1996) e REIS (2002). Por fim o descritor pode ser caracterizado como uma representação textual, utilizado, opcionalmente, na definição das atividades.

5.2 Atividade de Instanciação⁹

Nesta atividade, as tarefas são concebidas¹⁰, os ativos de processo são configurados. REIS (2002) salienta que as representações visual e textual do processo devem ser, totalmente, definidas nesta atividade, lembrando sempre que as perspectivas de fluxo de trabalho, fluxo de dados, recursos e habilidades, salientadas por DERNIAME et. al. (1999) e SOMMERVILLE (2003), devem estar presentes nesta representação. Para os autores, nesta atividade, o processo é instanciado e possui em sua concepção genética, os elementos ou moldes definidos na abstração. Os recursos (humano e ferramental) para execução do processo são alocados e o cronograma de implantação é definido. A atividade de instanciação também é responsável pela configuração da máquina de processo.

A máquina de processo é um conceito importante na configuração de processo de produção de software. Baseado neste fato, o texto apresenta a descrição formal de tal máquina.

A Máquina de Processo

SCHAEFER et. al. (1999) afirmam que a máquina de processo, pode ser classificada como um software com o objetivo de auxiliar na comunicação e coordenação das atividades realizadas pelos envolvidos no processo. Um dos

⁹ Atividade presente no modelo Arquitetônico de Reutilização de Processos proposto por FIORINI (2001) – vide capítulo 3.

¹⁰ A concepção das tarefas se dá a partir da divisão das atividades.

objetivos da máquina de processo é controlar a produção dos artefatos, classificados como componente de código ou componente de infra-estrutura (ambos classificados como ativos de processo). Sabendo que o objetivo de tal máquina é controlar a produção de componentes, faz-se necessário apresentar a teoria sobre o objeto em questão, o componente.

Segundo WANG (2000), os componentes são blocos de construção básicos, que alicerçam a engenharia de software baseada em componentes. A granularidade destes blocos implica na reusabilidade, produtividade e manutenibilidade dos componentes. No paradigma tradicional, um componente pode ser definido como inteiro, vetor, ponteiro ou uma biblioteca de funções. No paradigma orientado a objeto, um componente pode ser definido como classe, método ou conjunto de classes.

Já para YACOUN et. al. (2005), um componente de software é um conjunto de códigos que encapsula uma ou mais funcionalidades. Para os autores, os componentes de software têm duas descrições básicas: externa e interna. A descrição externa define, formalmente, a comunicação entre o componente e a plataforma por este utilizado. Já as características internas refletem aspectos sobre o procedimento ou processo de execução de um do componente.

Para DOUCET et. al (2002), um componente de software é uma unidade de composição que pode assumir a forma de uma função, de um objeto, de uma biblioteca ou de um programa completo. Os autores propõem a classificação dos componentes em níveis, segundo a sua abstração.

Este trabalho parte dos mesmos princípios propostos por WANG (2000) e DOUCET et. al (2002) e, também, classifica os componentes em níveis, conforme mostra a Figura 5.4, na qual é possível verificar a distribuição dos níveis “específico”, “comum” e “infra-estrutura” pelos eixos “dependência do domínio” e “granularidade” e constatar que quanto maior a granularidade, maior o grau de reutilização do componente. Por exemplo: os componentes classificados como “infra-estrutura”

fazem parte da composição da maioria dos softwares de uma determinada empresa. Por outro lado, os componentes “específicos” possuem um alto grau de “dependência do domínio”, conseqüentemente, baixo grau de reutilização.

Figura 5.4 - Níveis de abstração de um componente de software em relação aos domínios de aplicação (inferida a partir dos estudos de WANG (2000) e DOUCET et. al (2002))

Um fato de extrema importância, apresentado por DOUCET et. al. (2002), é a necessidade de integração dos componentes dada a sua granularidade. Os autores salientam que a integração de componentes pode ser feita de três formas:

- Programada: O “integrador de componentes” deve desenvolver um programa que componha dois ou mais componentes;
- Gráfica: A composição dos componentes se dá por meio de um aplicativo ou ferramenta. Integração esta que provê maior grau de automação no processo de composição e, conseqüentemente, maior produtividade;
- *Scrip*¹¹: Necessidade de desenvolvimento de um *script* para a composição de dois ou mais componentes.

Complementando a definição de componentes, proposta por DOUCET et. al. (2002), SEAMAN (1992) e BASILI et. al. (1992) relatam que um componente não pode ser classificado somente como código. Os autores deixam claro que os demais

¹¹ Um *script* pode ser caracterizado como um conjunto de linhas de comandos aglomeradas em um arquivo, tais comandos são interpretados pelo *shell* (interpretador) do sistema operacional

artefatos (documentos, questionários, algoritmos), também, podem ser classificados como tal.

Este trabalho compila a idéia de componentes, utilizando as definições citadas, da seguinte forma: um componente pode ser encarado como qualquer artefato gerado pelo processo fabril, sendo assim, tanto os fragmentos de código encapsulados para o reuso como os ativos de processo são classificados com tal título.

Um fator que irá interferir, diretamente, na organização da máquina de processo é a proposta de representação de um componente, elaborada por WANG (2000). O autor salienta que um componente de software deve ser representado por uma “*sextupla*” interativa:

$$C = \{Q, I, f, O, q0, F\} \quad (1)$$

onde C representa o componente em questão, Q traduz o conjunto de estados de um componente, I representa o conjunto de entradas, f provê a função de transformação de um componente, O denota o conjunto de saída, $q0$ apresenta o estado inicial e, por fim, F representa o estado final do componente.

Partindo da necessidade de armazenamento e recuperação de informações, e que todo componente tem um ciclo de vida definido, foi inferido, neste trabalho, um conjunto básico de informações, para configuração da máquina de processo, relativas:

- à gestação do componente (o projeto de concepção do componente);
- ao nascimento do componente (a implementação da versão 1);

- à evolução do componente (o crescimento), a qual também se dá pelo desenvolvimento das atividades de implementação e testes. Novas versões também configuram o crescimento do componente;
- à reprodução do componente (um componente “a” se unido a um componente “b” resulta em um componente “c”);
- à morte do componente: o componente não pode ser mais utilizado devido a evolução tecnológica que permeia o desenvolvimento do software.

Seguindo a mesma sistemática utilizada por WANG (2000), na definição dos componentes e com base no ciclo de vida apresentado, FABRI et. al. (2007a) propõem que uma máquina de processo seja estabelecida por meio de um sextupla.

$$Maq = \{G, N, E, R, EV, M\} \quad (2)$$

onde *Maq* armazena o nome da máquina de processo, *G* armazena informações sobre a gestação de um componente, *N* armazena informações relacionadas ao nascimento, *E* trata informações sobre a evolução, *R* está relacionada com a reprodução, *EV* armazena as informações dos envolvidos com a construção do componente e, por fim, *M* relata se um componente está em desuso ou morto.

Todas as variáveis, com exceção de *M* e *EV*, da “sextupla” *Maq* derivam novas tuplas, representadas a seguir:

$$G = \{i, p, C\} \quad (3)$$

onde *i* armazena as informações da instituição que direcionou a concepção de um determinado componente, *p* retrata as informações relacionados ao projeto para o qual o componente foi concebido e *C* apresenta as informações de um determinado componente (informações estas apresentadas por WANG (2000) – vide tupla (1)).

$$N = \{C, v, l\} \quad (4)$$

onde v representa a versão de um determinado componente, o nascimento se dá após o estabelecimento da versão 1 e l armazena o local de armazenamento do componente. É importante salientar que parte das informações do componente encontra-se na gestação e parte no nascimento, fato que levou a inserção de C na tupla N .

$$E = \{C, a, t\} \quad (5)$$

onde a armazena as informações relacionadas à atividade versus tempo, por exemplo: um componente levou 15 horas (tempo) na atividade de implementação. Já t armazena informações relativas o teste.

$$R = \{C, c, h\} \quad (6)$$

onde c armazena as informações de quais componentes foram compostos para a geração de um novo componente e h retrata a forma de composição de um componente que, segundo DOUCET et. al. (2002), pode ser classificada como programada, gráfica ou *script*.

Arquiteticamente, a máquina de processo possui uma estrutura aberta e deve ser configurada segundo as necessidades da empresa produtora de software usuária do modelo. Um exemplo de uma máquina configurada pode ser encontrada em FABRI et. al. (2007a).

Após formalizar o conceito da máquina de processo, o texto volta a descrever as atividades pertinentes ao modelo.

5.3 Atividade de Simulação¹²

Nesta atividade o processo gerado pelo modelo é testado em um ambiente simulado, permitindo prever problemas e estimar a duração do processo, quando este estiver em execução. Se problemas forem detectados nesta atividade, o engenheiro de processo deve retornar às fases anteriores para que novos ajustes sejam feitos. Nesta atividade, o conhecimento do processo deve ser distribuído na organização produtora de software.

5.4 Atividade de Execução

A atividade execução é desenvolvida sobre um processo de produção de software. Nesta atividade, a máquina de processo (gerada pelo modelo) entra em operação e provê a interação entre os envolvidos e o processo de produção de software. REIS (2002) prevê que a máquina de processo deva possuir características adaptativas, pois sua estrutura pode ser modificada, se o processo for adaptado a um ambiente não previsto na técnica de *primitivation*.

5.5 Atividade de Avaliação

A atividade de avaliação assume duas formas: estática e dinâmica.

A avaliação estática é executada após o término do processo. Seu principal objetivo é verificar as informações sobre o uso real das estratégias gerenciais adotadas pela organização durante a execução do processo.

A avaliação dinâmica é desenvolvida com o processo em execução, e seu principal objetivo é corrigir algum problema ou erro no processo, não percebido na atividade de simulação.

¹² A atividade de simulação foi inserida na proposta com base na atividade teste e integração do processo, presente no modelo 3C de HOLLENBANCH e FRAKES.

A atividade de avaliação possui forte relação com as idéias de generalização e avaliação de processos encerrados, apresentadas por REIS (2002). Em seu trabalho, o autor deixa explícito que as estratégias utilizadas na definição ou reutilização do processo devem ser avaliadas e, se necessário, generalizadas para compor os modelos de processos armazenados.

5.6 Atividade de Armazenamento

Atividade que tem como objetivo armazenar as informações do processo e os dados pertinentes à sua execução. Esta atividade está, visivelmente, relacionada à vertente de execução definida na forma de desenvolvimento (criação) do processo, proposta por REIS (2002) e FRANCH e RIBÓ (2002).

A Figura 5.5 apresenta o relacionamento das 6 atividades que compõem o modelo. Ao analisar a figura, é possível constatar que o modelo é representado por uma moenda, onde os requisitos são “despejados”, e as atividades de modelagem, instanciação e simulação são “alavancadas”, resultando em um processo executável, modelado com as perspectivas de fluxo de trabalho, fluxo de dados, recursos e habilidades. O processo é executado sobre um projeto, avaliado e, posteriormente, armazenado, juntamente, com a máquina de processo, em uma base de processos.

Figura 5.5 – Visão estendida do modelo

Além da base de processo, é perceptível na Figura 5.5 a presença de outras bases, tais como:

- Base de projetos (representada na Figura 5.2 pela sigla PRO) cujo objetivo é armazenar dados sobre os projetos executados a partir dos processos gerados pelo modelo. A base de projeto possui as seguintes informações na sua arquitetura:
 - produto de software gerado pelo projeto;
 - tamanho de produto gerado;
 - atividades desenvolvidas na execução do projeto;
 - artefatos gerados pelas atividades;

- habilidades necessárias para execução das atividades;
 - envolvidos no projeto (os envolvidos devem ser separados por atividade);
 - ferramentas utilizadas para automatização das atividades;
 - tempo de execução de cada atividade;
 - tempo de trabalho e retrabalho separado por envolvido no projeto;
 - falhas ocorridas no processo, e;
 - nível de satisfação do cliente do projeto em relação ao produto.
- Base estrutural da linguagem de modelagem de processo (representada na Figura 5.2 pela sigla PML) cuja finalidade é armazenar a estrutura sintática da PML;
 - Base de moldes (representada na Figura 5.2 pela sigla MO) para definição dos ativos de processo cujo objetivo é armazenar a configuração sintática para a definição dos ativos de processo. Sintaticamente, um ativo de processo deve possuir em sua estrutura as seguintes informações:
 - informações relativas ao projeto;
 - nome do ativo;
 - versão;
 - histórico da revisão;

- pessoas autorizadas a preencher tal ativo;
- introdução – deve fornecer uma visão geral de todo do ativo;
- finalidade;
- escopo – uma breve descrição do escopo do(s) projeto(s) ao(s) qual(is) ele está associado e tudo o que é afetado ou influenciado por este ativo;
- referências - apresenta uma lista completa de todos os documentos mencionados pelo ativo (ou artefato) de processo. O documento deve ser identificado por título, número de relatório (se aplicável), data e organização responsável pela publicação;
- visão geral – Esta subseção deve explicar como o documento está organizado;
- informações exclusivas do ativo – aqui o corpo do ativo de processo é modelado.

Alguns compartimentos que aparecem na Figura 5.5 não foram descritos, textualmente, entre eles destacam-se: as informações sobre projeto (classificados como requisitos); as atividades de seleção e adaptação. Tais compartimentos foram inseridos no modelo com o objetivo de aliar a visão de reuso de processo (quadrante 1 da Figura 5.1) à visão de meta-processo.

5.7 Requisitos Utilizados na Visão de Reuso de Processos

O compartimento de informações sobre projeto, classificado como requisito no modelo proposto, tem como objetivo direcionar a atividade de seleção de um determinado processo, armazenado na base de processos (PCS). A principal informação a ser “despejada” na moenda é a Especificação de Requisitos de

Software (ERS), artefato que define as características do produto a ser desenvolvido, os requisitos funcionais e não funcionais. De posse da ERS, o engenheiro de processo consulta a base de projetos (PRO) e verifica se existe algum projeto, já desenvolvido, que possua características semelhantes ao projeto solicitado – a atividade de seleção¹³ é executada. Caso a resposta seja positiva, o processo utilizado na execução de tal projeto é enviado para atividade de adaptação. Caso a resposta seja negativa o engenheiro seleciona o processo padrão (processo que pode ser executado em qualquer projeto), e adapta-o para o desenvolvimento do projeto.

5.8 Atividade de Adaptação (Visão de Reuso de Processos)

A atividade de adaptação¹⁴ proposta neste trabalho, possui características semelhantes à atividade de adaptação desenvolvida por HOLLENBANCH e FRAKES (1996). Para estes autores, o principal objetivo de tal atividade é adaptar o processo e reunir os requisitos específicos inerentes ao ambiente do projeto. No modelo proposto, nesta atividade o engenheiro de processo deve analisar as características ambientais e técnicas do projeto a ser desenvolvido, verificar a aderência do processo selecionado ao projeto adaptando, assim, as atividades e tarefas, se julgar necessário. Quando a adaptação for desenvolvida sobre processo padrão, o engenheiro de processo deve assegurar que ela seja replicada dentro da base de processo. O engenheiro de processo pode utilizar as informações, armazenadas nas bases de dados apresentadas na Figura 5.5, para a execução desta atividade.

A atividade de adaptação está, intimamente, relacionada ao procedimento de modelagem genérica de processo, ao nível de customização de projeto, mostrado por DERNIAME et. al. (1999), e à vertente abstrata definida na forma

¹³ A atividade seleção foi inserida na proposta com base no modelo definido por HENNINGER (1998) – vide capítulo 4.

¹⁴ A atividade de adaptação também é citada nos trabalhos de HOLLENBANCH e FRAKES (1996) – modelo 3C BECKER et. al. (1997) – modelo (BHV), SUCCI et. al. (1997) – modelo Gertude, HENNINGER (1998) – modelo (H), REIS (2002) e FRANCH e RIBÓ (2002).

desenvolvimento do processo, proposta por REIS (2002) e FRANCH e RIBÓ (2002). Em seus trabalhos, os autores apresentam a configuração de um processo a partir de uma visão abstrata, classificada como uma estrutura de alto nível, cuja função é subsidiar a criação de elementos reguladores para a definição, customização e organização das atividades do processo. Neste trabalho, tal configuração é caracterizada como processo padrão.

5.9 – Considerações Finais em Relação ao Modelo Preliminar

Como pôde ser verificado neste capítulo, existe um forte relacionamento entre a proposta efetuada e os modelos 3C de HOLLENBANCH e FRAKES; Gertude de SUCCI, BENEDICENTI, PREDONZANI e VERNAZZA; (BHV) de BECKER, HAMANN E VERLAGE; (H) de HENNINGER e o (F) Modelo Arquitetônico de Reutilização de Processos de FIORINI (vide Tabela 5.1).

Visão 1 – VISÃO DE REUSO		Visão 2 – VISÃO DE DEFINIÇÃO	
Modelo	Literatura	Modelo	Literatura
Seleção	(H)	Requisitos	3C.
Adaptação	3C, BHV, (H).	Modelagem	3C, Gertude, BHV, (F)
		Instanciação	(F)
		Simulação	3C
Visão 3 – VISÃO COMPARTILHADA			
	Modelo	Literatura	
	Execução	Gertude. BHV	
	Avaliação	BHV	
	Armazenamento	(H)	

Tabela 5.1 – Visões: das atividades para criação e organização de um processo do modelo 3C de HOLLENBANCH e FRAKES e da proposta.

A Tabela 5.1 é confeccionada seguindo a idéia de visões apresentadas na Figura 5.2, fato este perceptível ao analisar a posição dos quadrantes de figura e da tabela.

Por fim, é importante salientar que as atividades “definir requisitos do processo”, “modelar processo” e “simular processo”, apresentadas no modelo 3C, são atividades utilizadas para a definição de um processo de produção de software,

fato este que levou à inserção delas no modelo. Já a atividade de instanciação foi inserida devido a sua definição, apresentada por REIS (2002). Recordando que o objetivo desta atividade, no modelo, é partir de um processo abstrato, modelado segundo os requisitos levantados, e conceber a máquina de processo.

Capítulo 6 – Método de Pesquisa Utilizado no Desenvolvimento do Trabalho

Antes de iniciar a contextualização do método de pesquisa utilizado neste trabalho, é válido recordar que os capítulos 1, 2, 3, 4 e 5 demonstram a motivação, o embasamento, e a proposta preliminar da tese. A seção 2.1, o capítulo 7, juntamente, com os anexos A e B apresentam a organização dos processos das fábricas brasileiras e estrangeiras. Por fim, os capítulos 8, 9 e 10 espelham as contribuições deste trabalho junto à comunidade acadêmica.

A literatura científica mostra que, ao longo dos tempos, a humanidade, num processo lento, reuniu extensas informações que foram traduzidas como conhecimentos. A necessidade levou o ser humano a observar o seu habitat natural e desenvolver utensílios simples, facilitando assim, suas atividades cotidianas. Atualmente, a constante busca por um desenvolvimento tecnológico, que proporcione maior facilidade para realização de tarefas, está presente em vários campos do conhecimento. Sendo assim, o progresso científico é produto desta busca constante, pela qual o homem procura explicar e desenvolver inferências sobre os objetos que o cercam, com o objetivo de promover novas descobertas.

Para FABRI et. al. (2005a), a evolução científica de uma área de pesquisa é alicerçada por três ações que estão, intimamente, ligadas: pesquisar, ensinar e apresentar os resultados da pesquisa (a extensão) – vide Figura 6.1.

Figura 6.1 – Processo de evolução científica (retirado de FABRI et. al. (2005a) página 70).

Analisando a Figura 6.1, é possível verificar que a humanidade possui novas necessidades em várias áreas do conhecimento, estas, por sua vez, promovem o desenvolvimento da pesquisa. O ato de pesquisar deve ser alicerçado por um processo científico (como fazer ciência e gerar novos conhecimentos?). O repasse dos conhecimentos desenvolvidos na fase de pesquisa atinge a fase do ensino. O ato de ensinar tem como objetivo a formação de novos profissionais e cientistas que busquem soluções para diversos problemas (aplicação do conhecimento). A fase de extensão tem como pretensão, a apresentação dos resultados à humanidade, fechando assim, o ciclo evolutivo do processo científico.

Assim, toda e qualquer pesquisa científica deve trazer consigo um valor agregado, para que a humanidade possa se beneficiar e conquistar uma melhoria constante em sua qualidade de vida (FABRI et. al. (2005a)). Os autores, anteriormente citados, ainda, definem que o ato de pesquisar pode ser ilustrado por meio da Figura 6.2. A pesquisa é regida pela transformação da matéria prima (informação, conhecimento, ciência), efetuada pelo pesquisador (mão de obra) que utiliza recursos (financeiros, bibliotecas) resultando em um produto ou serviço que traz um valor agregado para o cliente (humanidade). A transformação é permeada por um método de pesquisa.

Figura 6.2 - O ato de pesquisar (retirado de FABRI et. al. (2005a) página 73)

Em seu trabalho, os autores propõem alguns passos para compor um projeto de pesquisa¹. Tais passos foram elucidados no capítulo 1 deste trabalho e, para facilitar a leitura, também serão apresentados a seguir:

- Passo 1 - definição da questão de estudo: a questão de estudo direciona a pesquisa em si, estabelecendo assim, o foco na qual o pesquisador irá imergir. A seleção de uma questão demanda conhecimentos sobre a área da ciência na qual se deseja navegar. A percepção das necessidades da humanidade, também, pode ajudar na definição da questão de estudo;
- Passo 2 - estabelecimentos das proposições: uma proposição direciona algum fato que deva ser examinado dentro da pesquisa. Ressalta-se que uma proposição pode ser confirmada ou não, após a realização da pesquisa;
- Passo 3 - definição do método de pesquisa: a definição do método de pesquisa está ligada à questão que se deseja responder. Um exemplo de definição do método de pesquisa, baseado na forma da questão, é definido por YIN (2005) - Tabela 6.1. O autor ressalta, ainda, que a escolha do método está relacionada a duas condições básicas: controle que o pesquisador possui sobre os eventos comportamentais efetivos e o foco em fenômenos históricos, em oposição a fenômenos contemporâneos;

¹ A Idéia de dividir a composição de projeto de pesquisa em passos, também, é compartilhada por YIN (2005).

Estratégias	Forma da questão de pesquisa	Exige controle sobre eventos comportamentais	Focaliza acontecimentos contemporâneos
Experimento	Como, por que	Sim	Sim
Levantamento	Quem, o que, onde, quantos, quanto	Não	Sim
Análise de arquivo	Quem, o que, onde, quantos, quanto	Não	Sim/Não
Pesquisa histórica	Como, por que	Não	Não
Estudo de Caso	Como, por que	Não	Sim

Tabela 6.1 – Situações relevantes para diferentes estratégias de pesquisa (retirada de YIN, página 24)

- Passo 4 - delinear as unidades de análises: Quais serão as unidades (experimentos, casos, simulações) que devem ser mapeadas e analisadas?
- Passo 5 - Relacionar os dados gerados nas unidades de análise e interpretar as constatações: De posse dos dados gerados nas unidades de análise o pesquisador deve inferir e desenvolver constatações, contribuindo assim, com a evolução científica.

Na composição do projeto de pesquisa deste trabalho, serão seguidos os passos apresentados por FABRI et. al. (2005a) e YIN (2005). Sendo assim, é pertinente afirmar que a questão a ser respondida neste trabalho é: É possível construir um modelo que possibilite a criação ou organização de um processo de software com características fabris?

A necessidade da criação e/ou organização de um processo de produção de software é uma necessidade constante no contexto nacional e mundial. Esta informação está alicerçada na análise numérica sobre o mercado produtivo de software apresentado no capítulo 1 deste trabalho.

De posse da questão de pesquisa, faz-se necessário definir as proposições, que serão utilizadas como fatores de direcionamento do trabalho. As proposições, definidas neste trabalho estão, intimamente, ligadas a questão de pesquisa e a

teoria de fábrica de software e processo de software. A seguir são enumeradas as três proposições delineadas neste trabalho:

1. O processo de produção de uma fábrica de software é definido por meio de uma estrutura formal (métodos, procedimentos que levem a estruturação do processo de produção). Como pode ser verificado no capítulo 4, existem vários modelos, métodos, teorias que auxiliam uma empresa produtora de software a criar ou organizar um processo de produção de software. Estes métodos são aplicados por tais empresas? Se sim, como eles são aplicados, quais os resultados gerados com tal aplicação, eles podem alterar o modelo preliminar proposto no capítulo 5.
2. Existe um processo genérico de produção de software que pode ser aplicado ao contexto fabril. A instanciação de um processo fabril genérico, após efetuar uma análise das empresas, caracterizadas como fábrica de software, pode auxiliar na concepção do modelo, principalmente, na visão de reuso de processo de software, apresentada na proposta preliminar. Este processo pode ser caracterizado com o primeiro processo inserido na base de processos da proposta efetuada no capítulo 5 e posteriormente ser reutilizado.
3. O ciclo de produção da uma fábrica de software é classificado como curto, médio e/ou longo. A definição do ciclo de produção de uma fábrica de software pode interferir na utilização do modelo, principalmente, por exemplo: o tempo de definição de uma fábrica de software de ciclo longo pode ser mais longo que o tempo de definição de uma fábrica de software de ciclo curto, pois a primeira possui um conjunto maior de atividades a serem modeladas, instanciadas, simuladas e executadas se comparada com a segunda.

Ao analisar as três proposições é possível perceber que todas possuem relação direta com o levantamento bibliográfico sobre o tema, apresentado até o momento, e com a proposta preliminar. Este fato é comprovado ao analisar as palavras grifadas nas justificativas elaboradas para cada proposição.

Seguindo os passos para definição de um projeto de pesquisa proposto por FABRI et. al. (2005a), a próxima etapa a ser percorrida é a definição do método de pesquisa (passo 3). O método utilizado para o desenvolvimento deste trabalho é o estudo de caso. A escolha motivou-se a partir dos seguintes fatos: o pesquisador não tem o controle sobre eventos que regem a pesquisa e o tema central deste trabalho configura-se como um fenômeno contemporâneo real, pois conforme citado no capítulo 2, o conceito de fábrica de software está, intimamente, relacionado à idéia de processo de produção.

Além dos fatos apresentados, existe no Brasil um pequeno grupo de empresas que desenvolvem software com base em modelos da qualidade e produtividade, em contra partida, um grande grupo de empresas encontra dificuldade em produzir software em escala industrial. Fato este que justifica examinar de maneira sistematizada como as organizações, inseridas no primeiro grupo, se comportam. É somente com a utilização do método de pesquisa baseado em estudo de caso que isto se torna possível.

A proposta de realização do estudo de caso se dará em duas etapas. Na etapa 1, serão visitadas as 6 empresas caracterizadas como fábrica de software cuja certificação de qualidade esteja comprovada (vide Tabela 6.2). Na segunda será relatada a aplicação do modelo proposto no capítulo 5. Esta aplicação foi realizada por uma empresa na definição do processo de produção de software de suas parceiras.

Definida a questão da pesquisa, delimitadas as proposições e estabelecido o método de pesquisa, faz-se necessário, neste momento, planejar o desenvolvimento, a execução e a análise do estudo de caso.

Segundo YIN (2005), no planejamento do estudo de caso, é necessário percorrer as seguintes etapas:

- Etapa 1: A pesquisa de campo irá envolver um ou mais casos: O trabalho proposto irá realizar estudos de casos múltiplos, pois o mesmo não está sendo desenvolvido sob uma circunstância exclusiva e os casos não representam um teste crucial à teoria existente;

Casos Desenvolvidos em Campo						Casos de Literatura (clássicos)	
Empresas Brasileiras	Justificativa para escolha do caso ²	Início das Negociações	Data de visita as empresas	Entrevistados	Tempo utilizado nas entrevistas	Empresa	Referencial bibliográfico.
<u>EMPRESA A</u>	CMMI ³ – 3 ⁴	Abril/2005	Agosto/2005	Diretor da produção da fábrica de software. Gerente da qualidade (SQA). Equalizador. Líder de Equipe. Coordenador da Célula de Testes.	20 horas	System Development Corporation (SDC) (EUA) Hitachi (JP) Toshiba (JP) NEC (JP) Fujitsu (JP)	CUSUMANO (1991)
<u>EMPRESA B</u>	CMMI – 3	Mai/2005	Setembro/2005	Gerente da Qualidade (SQA)	6 horas		
<u>EMPRESA C</u>	CMMI – 2	Julho/2005	Novembro/2005	Gerente da Qualidade (SQA)	7 horas		
<u>EMPRESA D</u>	CMMI – 5	Novembro/2005	Janeiro/2006	Gerente da Qualidade (SQA)	5 horas		
<u>EMPRESA E</u>	ISO 9001	Janeiro/2006	Fevereiro/2006	Diretor Geral. Membro do grupo de garantia da qualidade.	8 horas		
<u>EMPRESA F</u>	CMMI – 2	Fevereiro/2006	Março/2006	Gerente da Qualidade (SQA). Gerente de Produção	6 horas		
<u>EMPRESA X</u>	CMMI – 2	Janeiro/2006 Início da aplicação do modelo	Novembro de 2006				

Tabela 6.2 – Lista de casos apresentados neste trabalho. EUA – Estados Unidos de América. JP – Japão. A coluna início das negociações indica o primeiro contato estabelecido com as empresas para a realização das visitas.

- Etapa 2: Definir as unidades de análises: Quantos e quais serão os casos estudados? Neste trabalho serão apresentados seis casos, empresas com atividade relacionada à fabricação de software⁵. Além destes seis, foram apresentados, no capítulo 2, outros cinco, advindos da literatura, denominado,

² O critério para escolha está relacionado com a condição da organização possuir um sistema da qualidade definido.

³ Capability Maturity Model Integration. O número a frente da sigla traz o nível de maturidade da empresa. (CMMI-SE/SW (2002)).

⁴ Em agosto de 2005, época em que o estudo de caso foi desenvolvido, a EMPRESA A possuía certificação CMMI nível 2.

⁵ Os nomes das empresas que cederam as informações para realização de tal estudo não serão divulgados, pois o autor deste trabalho não possui autorização formal para isto.

neste trabalho como casos clássicos⁶. A Tabela 6.2 apresenta relação dos casos. Ressalta-se que a empresa que se propôs a aplicar o modelo apresentado no capítulo 5, também é considerada um caso (EMPRESA X), caracterizando assim o décimo segundo caso. Segundo YIN (2005), a quantidade de casos apresentada neste trabalho está dentro dos padrões aceitáveis;

- Etapa 3: Estruturar o protocolo para realização do estudo de caso: O protocolo tem como objetivo auxiliar o pesquisador na coleta dos dados. YIN (2005) salienta que o protocolo é uma ferramenta essencial para aumentar a confiabilidade do estudo de caso e destina-se a orientar o pesquisador na condução ou execução da pesquisa de campo. O protocolo para o desenvolvimento da etapa 1 do estudo de caso, utilizado neste trabalho, é apresentado na Tabela 6.3. Já o protocolo utilizado para o desenvolvimento da etapa 2 pode ser verificado na Tabela 6.4. A parte esquerda de ambas as tabelas apresenta os pontos direcionadores, para o desenvolvimento do protocolo, propostos por YIN (2005). Já a parte direita, apresenta o protocolo desenvolvido e aplicado nas empresas, para execução do estudo. É importante salientar que para a realização da pesquisa, o protocolo proposto para execução da etapa 1, foi analisado, previamente, pelas empresas e, posteriormente, as visitas foram agendadas e o estudo de caso efetuado. Já o protocolo proposto para a etapa 2, foi construído pelo autor deste trabalho com base no modelo preliminar apresentado no capítulo 5 e este não foi analisado pela EMPRESA X.

Analisando a Tabela 6.3 é possível estabelecer uma relação do roteiro para o desenvolvimento do estudo do caso com as teorias apresentadas nos capítulos 2, 3, 4 e 5. Esta relação pode ser verificada, por meio da Figura 6.3, na qual é possível perceber a presença do(s)(a)(as):

⁶ O autor deste trabalho analisou também o processo de produção de software da Microsoft (CUSUMANO e SELBY (1997)). Porém tal processo não foi inserido na estrutura textual, pois ele não é considerado fabril. Ressalta-se, também, que os casos receberam este título devido a riqueza de alguns detalhes presente em sua estrutura descritiva.

	Estrutura de Protocolo (adaptada de YIN (2005)).	Protocolo utilizado no desenvolvimento da etapa 1.
Uma seção introdutória, com a finalidade de apresentar os objetivos do protocolo as empresas, também, foi desenvolvida. Esta seção não é apresentada nesta tabela, seu desenvolvimento foi motivado como meio de introduzir o protocolo junto a empresas.		
A	Questões e proposições do estudo de caso:	
	A1	Questão: É possível construir um modelo que possibilite a criação ou organização de um processo de software com características fabris?
	A2	Proposições: O processo de produção de uma fábrica de software é definido por meio de uma estrutura formal (métodos, procedimentos que levem a estruturação do processo de produção). Existe um processo genérico de produção de software que pode ser aplicado ao contexto fabril. O ciclo de produção da uma fábrica de software é classificado como curto, médio e/ou longo.
B	Estrutura teórica para desenvolvimento do estudo de caso.	Referencial bibliográfico sobre fábrica de software apresentado no capítulo 2. Nota: O protocolo enviado para as empresas possuía em sua estrutura as principais definições sobre fábrica de software, condensadas em uma página.
C	Procedimentos de coleta (a execução):	
	C1	Locais a serem visitados.
	C2	Pessoas de contato nas empresas.
	C3	Lista de documentos analisados.
D	Roteiro para o desenvolvimento do estudo de caso.	O roteiro para o desenvolvimento do estudo de caso está dividido em 7 visões (Grande parte dos itens inseridos neste roteiro, também, está presente na obra de CUSUMANO (1991):
		1 – Caracterização da empresa (descrição geral da empresa pesquisada).
		2 - Visão sob a ótica dos insumos: Define o processo de produção da fábrica de software como: <ul style="list-style-type: none"> • Curto; • Médio; • Longo.
		3 – Visão das atividades do processo de produção: <ul style="list-style-type: none"> • Atividades; • Modelo de processo (cascata, incremental, ...); • Envolvidos com as atividades do processo; • Características (Leve ou Pesado).
		4 – Visão das atividades relacionadas a gestão operacional no contexto fabril para software: <ul style="list-style-type: none"> • Gestão de Qualificação; • Gestão de Configuração; • Gestão de Projetos: <ul style="list-style-type: none"> o Estabelecimento de cronogramas; o Definição de recursos; o Metrificação (por produtividade, por pessoa): <ul style="list-style-type: none"> • Média por programador; • Números de defeitos no projeto; • Controle temporal de cada atividade; • Formalização das métricas (por exemplo: utilização de análise por pontos de função).
		Referencial Bibliográfico norteador de visão. FABRI et. al. (2007) e (2007b) FERNANDES E TEIXEIRA (2004) LI et. al (2001) PAULA FILHO (2003) PRESSMAN (2002) ROCHA (2004) SOMMERVILLE (2003) SLACK (2002) BEMER (1969) BASILI et. al. (1992) CANTONE (1992) FERNSTROM et. al. (1992) ROCKWELL e GERA (1993) RUHE (1999) LI et. al (2001) FABRI et. al. (2004, 2004a, 2004b) FERNSTROM e TEIXEIRA (2004)

		<p>5 - Visão sob a ótica de armazenamento de dados (ferramentas, componentes, dados):</p> <p>5.1 – Ferramentas:</p> <ul style="list-style-type: none">• Desenvolvimento (ambientes de desenvolvimento, por exemplo: Eclipse para Java);• Teste (geradores de casos de testes);• Gerador de código;• Gestão de configuração e de versões;• Máquina de processo. <p>5.2 – Biblioteca de componentes:</p> <ul style="list-style-type: none">• Configurada e organizada;• Forma de acesso;• Aspecto de classificação:<ul style="list-style-type: none">o Infra-estrutura;o Comum;o Específica. <p>5.3 – Organização da base de dados:</p> <ul style="list-style-type: none">• Estruturação dos dados relacionados ao planejamento e controle do projetos.	THORESON (1989) LI et. al (2001) TRINDADE (2006) FABRI et. al. (2004, 2004a, 2004b)
		<p>6 – Análise sob a ótica dos produtos gerados:</p> <ul style="list-style-type: none">• Orientação a domínio (a empresa desenvolve software em um domínio do conhecimento apenas);• Orientação a produto (a empresa desenvolve somente um ou dois produtos);• Orientação à tecnologia (a empresa desenvolve produtos somente uma ou duas plataformas).	FABRI et. al. (2005)
		<p>7 – Forma de criação e organização do processo.</p>	HOLLENBANCH e FRAKES (1996) BECKER et. al. (1997) SUCCI et. al. (1997) HENNINGER (1998) REIS (2002). FRANCH e RIBÓ (2002).
Análise e Validação do Estudo de Caso			
E	E1	Validade do Constructo.	<p>Análise das fontes de evidências:</p> <p>As evidências foram coletadas nas entrevistas e na análise dos documentos do processo. Tais evidências foram comparadas e se inconsistências fossem encontradas, elas eram esclarecidas junto a um terceiro entrevistado.</p> <p>Análise do relatório do estudo de caso:</p> <p>Os entrevistados validaram os relatórios dos estudos, logo após a sua concepção.</p> <p>Os relatórios dos estudos foram configurados como artigos técnicos, os envolvidos nas entrevistas participaram da co-autoria dos artigos.</p>
	E2	Validade interna	Relacionamento das informações obtidas nos estudos de caso com a teoria sobre fábrica de software, processo de produção de software, reuso de software.
	E3	Validade externa	Os artigos técnicos foram submetidos a conferências e congressos da área, alguns foram aceitos como é caso de FABRI et. al. (2006b). FABRI et. al. (2007, 2007b, 2007c e 2007d), e publicados.

Tabela 6.3 – Protocolo utilizado para a execução da etapa 1 do estudo de caso.

- modelo de classificação por níveis, apresentado por TRINDADE (2006): Este modelo é relacionado com os modelos fabris apresentados no capítulo 2 (vide Figuras 2.3, 2.4, 2.9, 2.10, 2.11);
- insumos do processo de produção: Os insumos definem se o processo fabril está organizado como curto, médio ou longo, conceitos estes definidos no capítulo 2, por exemplo: se uma fábrica tem como entrada ordens de serviços e

especificações do projeto de software, ela é classificada sob a ótica do ciclo curto (fábrica de código);

	Estrutura de Protocolo (adaptada de YIN (2005)).			
Não foi desenvolvida uma seção introdutória, pois o protocolo utilizado para a execução da etapa 2 não foi apresentado a <u>EMPRESA X</u> .				
A	Questões e proposições do estudo de caso:		Idem ao protocolo da Etapa 1.	
	A1	Questão:	Idem ao protocolo da Etapa 1.	
	A2	Proposições:	Idem ao protocolo da Etapa 1.	
B	Estrutura teórica para desenvolvimento do estudo de caso.		O modelo preliminar apresentado no capítulo 5 foi utilizado como estrutura teórica para o desenvolvimento do estudo de caso	
C	Procedimento de coleta dos dados			
	C1	Local a serem visitados	Área de produção da <u>EMPRESA X</u> . Área de produção da fábrica de software criada pela <u>EMPRESA X</u> , a Fábrica de Software da Faculdade de Tecnologia de Ourinhos.	
	C2	Pessoas de contato nesta etapa.	Gerentes de projeto da <u>EMPRESA X</u> . Responsável para transferência das informações relacionadas ao projeto. Engenheiro de Processo, pessoa que utilizou o modelo para a criação e organização do processo de produção de software.	
	C3	Documento analisados nesta etapa.	Todos os documentos gerados com a execução do modelo preliminar proposto no capítulo.	
D	Roteiro para desenvolvimento do estudo de caso nesta etapa.		Histórico e descrição do ambiente para realização do estudo de caso.	Mecanismo proposto no capítulo 5 deu a tônica para a definição do roteiro.
			Levantamento de requisitos para a definição do processo de produção de software.	
			A atividade de modelagem proposta no modelo preliminar.	
			A atividade de instanciamento proposta no modelo preliminar.	
			A atividade de simulação proposta no modelo preliminar.	
			A atividade de execução proposta no modelo preliminar.	
			A atividade de avaliação proposta no modelo preliminar.	
			As Atividades de armazenamento, seleção e adaptação propostas no modelo preliminar.	
E	Análise e Validação do Estudo de Caso			
	E1	Validade do Constructo.	Análise das fontes de evidências: Os dados gerados com a execução do modelo foram coletados durante o ano de 2006. As evidências foram comparadas e algumas inconsistências também foram encontradas, elas proporcionaram uma evolução do modelo.	
	E2	Validade interna	Não foi aplicado, visto que a execução do modelo tem como objetivo responder a questão de pesquisa.	
	E3	Validade externa	Será utilizada a mesma sistemática utilizada na etapa 1. Os artigos técnicos encontram-se em fase de desenvolvimento.	

Tabela 6.4 - Protocolo utilizado para a execução da etapa 2 do estudo de caso.

Figura 6.3 - Modelo utilizado na composição do roteiro para o desenvolvimento estudo de caso (Etapa 1)

- processo de produção de software que deve ser configurado segundo os apresentado no capítulo 4 (cascata, incremental, orientado a reuso, espiral, entre outros). Além do modelo, informações das atividades e tarefas que compõem um processo de software (vide Figura 4.10) devem ser mapeadas nos casos. A caracterização da execução do processo em rápido/leve (processos de produção de software que tendem ao *eXtreme Program* (XP⁷)) ou lentos/pesados (processos de produção de software que tendem ao *Unified Process* (UP)) são mapeados no roteiro. Esta caracterização pode ser utilizada como guia para auxiliar a construção de processos e deve ser mapeada nos casos. Um estudo sobre tal caracterização aplicada em fábricas de software foi desenvolvido por ROCHA et. al. (2004);
- bases de ferramentas, componentes e dados presente no processo fabril (vide LI et. al. (2001) - Figura 2.2, FABRI et. al. (2004, 2004a e 2004b) - Figura 3.6, e TRINDADE (2006) - Figura 3.8). O conceito de armazenamento de informações ou base de dados também é uma constante na teoria de reuso de processo, esta afirmação pode ser comprovada ao analisar os estudos efetuados por BECKER et. al. (1997), SUCCI et. al. (1997) e FIORINI (2001);
- produtos gerados pelo processo de produção de software. Visão esta que estabelece se a fábrica de software está orientada a domínio, conforme os estudos efetuados por FABRI et. al. (2005). A idéia de orientação a domínio também é compartilhada por FENANDES e TEIXEIRA (2004) nas Figura 2.10 e 2.11;
- e por fim, visão administrativa operacional, ligada a questão de gestão de projetos de software, conceito este importante dentro de um ambiente fabril de produção de software. Tal importância é destaca nas definições de fábrica de

⁷ Segundo GRANVILLE e DAVE (2002), *eXtreme Program* é uma metodologia ágil para equipes de desenvolvedores de software caracterizadas como pequenas e médias. O diferencial da metodologia está no suporte a constante mudança de requisitos.

software estabelecidas por BEMER (1969)⁸, CUSMANO (1989), CUSUMANO (1991), RUHE (1999), LI et. al. (2001), FERNANDES e TEIXEIRA (2004), FABRI et. al. (2004, 2004a e 2004b) e por TRINDADE (2006);

De posse do protocolo – etapa 1, a próxima fase a ser percorrida é a execução. Nesta fase foram executadas as entrevistas, analisados os documentos do processo, disponibilizados pelas empresas, e os relatos das informações colhidas foram descritas no formato de um relatório e de um artigo técnico.

Cabe ressaltar que todas as entrevistas foram gravadas em fitas K7. Já a reprodução dos documentos relacionados ao processo de produção de software não foi autorizada por parte das empresas.

A validade das informações mapeadas nos estudos de caso foi efetuada através do desenvolvimento e submissão de artigos técnicos a congressos da área. Os artigos foram (ou estão no caso da aplicação da Etapa 2) escritos segundo a técnica apresentada na Figura 6.4. Em tal figura, um fato (conceito inerente à fábrica de software) é descrito sob a ótica das entrevistas, da análise de documentos e da observação direta do ambiente de produção.

Já para a etapa 2, todas as informações geradas com a execução do modelo foram descritas pelo o autor deste trabalho. A descrição completa do caso da EMPRESA X pode ser verificada no capítulo 10 do trabalho.

⁸Autor trata, principalmente, questões de medidas de produtividade e qualidade, registros financeiros mantidos por custo da programação e gerenciamento que dê subsídios à previsão e à estimativa de dados de futuros projetos.

Figura 6.4 – Forma de descrição dos fatos mapeados nos estudos de casos

A relação dos dados gerados nas unidades de análise (EMPRESAS de A a E) serão comparadas com os casos estrangeiros, apresentados no capítulo 2 (vide capítulo 8).

Por fim, o capítulo 9 apresenta a contribuição dos casos para com o modelo proposto no capítulo 5.

Capítulo 7 – As Fábricas Brasileiras

Este capítulo tem como objetivo, apresentar um relato condensado das seis fábricas brasileiras que participaram da pesquisa de campo. A estruturação deste capítulo respeitará o roteiro utilizado para execução do estudo de caso, delineado na parte D da Tabela 6.3. Salienta-se que uma descrição detalhada sobre as fábricas brasileiras pode ser encontrada no anexo A deste trabalho.

7.1 – Caracterização das Empresas

A EMPRESA A foi fundada em 1995 e, atualmente, possui 1200 colaboradores. Seu principal objetivo é prover soluções de TI para o mercado brasileiro. A empresa em questão atua, diretamente, nas áreas de consultoria de TI e desenvolvimento de software.

Fundada em 1970 a EMPRESA B, desenvolve soluções de TI para o mercado nacional e internacional¹. Seu foco de atuação está alicerçado em sustentação e integração de sistemas, transferência e recepção de tecnologia, *help desk* e produção de software. Atualmente, a empresa conta com 4.000 colaboradores atuando no Brasil. Salienta-se que uma pequena parte destes colaboradores atua no mercado de TI norte-americano.

Já a EMPRESA C foi criada em 1989 e seu principal foco é prover soluções de TI para o mercado brasileiro. Para isto conta com 8000 colaboradores espalhados por vários estados. Suas soluções mercadológicas prevêm consultoria na área de engenharia de processos, desenvolvimento de software, manutenção de sistemas e gestão de ambiente e infra-estrutura.

¹ A EMPRESA B pode ser encarada como uma multinacional brasileira.

A EMPRESA D foi criada em 1964 e é caracterizada como uma multinacional estrangeira. No Brasil ela possui sede no Rio de Janeiro (unidade visitada), São Paulo, Araraquara e Florianópolis. O principal objetivo de tal empresa é prover soluções de TI para o mercado internacional. Atuando nas áreas de consultoria de negócios, provimento de infra-estrutura para telecomunicações, desenvolvimento de software e gerenciamento de processos que não sejam o *core business* de seus clientes. Em âmbito mundial a empresa conta com 120.000 colaboradores (2.000 trabalhando, ativamente, no mercado brasileiro). Destes, cerca de 33.000 exercem o cargo de engenheiro de software e a sua carteira de clientes chega a 35.000 clientes.

A EMPRESA E foi fundada em 1990 e atua em um âmbito regionalizado, atendendo, principalmente, os estados de São Paulo, Paraná e Santa Catarina. Focada no desenvolvimento de soluções de TI que envolvam a utilização de sistemas ERP (*Enterprise Resource Planning*), a empresa conta, atualmente, com cerca de 40 colaboradores.

Por fim, a EMPRESA F foi fundada em 1996 e provê soluções de TI nas áreas de comunicações de dados, desenvolvimento de softwares para internet e consultoria em processos de negócios. Salienta-se que cerca de 600 colaboradores atuam, diretamente, nas áreas destacas neste parágrafo.

7.2 – Insumos e Atividades do Processo de Produção de Software

Conforme relatado no capítulo 2 os insumos do processo definem se o processo fabril está organizado como curto, médio ou longo. Nas empresas participantes da pesquisa de campo foi possível constatar que todas elas, excluindo a D, são norteadas pelas atividades pertinentes ao ciclo longo. Um fato que chama a atenção nos estudos realizados (excluindo a EMPRESA E), é que 5 empresas (A, B, C, D e F) definem unidades organizacionais de produção e teste de código, e as denominam como fábrica de software. Estas unidades são contratadas, na maioria das vezes, internamente, pelas unidades responsáveis pelo desenvolvimento das

atividades de análise de negócio e projeto de software, e estas, por sua vez, são contratadas por clientes que necessitam que algum tipo de software seja desenvolvido.

Em relação às atividades do projeto é possível verificar que:

- EMPRESA A realiza as atividades de levantamento de requisitos, modelagem de negócio², projeto de software, equalização, codificação, teste (unitário e integrado) e entrega (envolvendo os conceitos de manutenção, homologação e implantação de software). A unidade organizacional delineada como fábrica de software é responsável pela equalização, codificação, teste e entrega do código;
- EMPRESA B realiza as atividades de modelagem de negócio, projeto de software, equalização, codificação, teste (unitário, integrado e de aceitação), manutenção corretiva e melhorias. A unidade organizacional delineada como fábrica de software é responsável pela equalização, codificação, teste e entrega do código;
- EMPRESA C realiza as atividades de levantamento de requisitos (veja seção A.3.4) modelagem de negócio, projeto de software, equalização, codificação e teste (unitário, integrado e de aceitação). A atividade de entrega nesta empresa é chamada de implantação. A unidade organizacional delineada como fábrica código de software é responsável pela execução das atividades de equalização e codificação. Uma outra unidade organizacional denominada como fábrica de teste é responsável pelo teste unitário e integrado;
- EMPRESA D realiza as atividades de levantamento de requisitos e projeto de software em algumas unidades e repassam as atividades codificação e teste (unitário, integrado e aceitação) e entrega do código para outras, caracterizando,

² Salienta-se que a atividade de modelagem de negócio, também, é chamada de análise de sistemas em algumas empresas.

assim, o conceito de *outsourcing*. Salienta-se que a atividade de equalização também é praticada na unidade visitada;

- EMPRESA E realiza as atividades de coleta de requisitos, modelagem de negócio, projeto de software, equalização, produção de código, teste³, implantação (envolvendo o conceito de manutenção de software) e revisão (vide capítulo 2);
- EMPRESA F realiza as atividades de levantamento de requisitos, análise de sistemas, projeto de software, equalização, codificação, teste (unitário e integrado), entrega e manutenção. A unidade organizacional delineada como fábrica de software é responsável pela equalização, codificação, teste e entrega do código.

Em relação ao modelo de processo, foi possível detectar que todas as empresas, excluindo a F, utilizam a sistemática incremental. A empresa destacada neste parágrafo utiliza um modelo misto, que engloba duas sistemáticas: evolucionária, focado na técnica de prototipação, e incremental.

Apresentado o modelo e as atividades do processo, a parte D da Tabela 6.3 prevê que sejam destacados os envolvidos com o processo de produção das empresas.

Salienta-se que o mapeamento dos envolvidos privilegiou a visão processual da organização, delineada na Figura 2.3. Porém, alguns envolvidos com a visão gerencial, também, foram citados pelos entrevistados, e estes não foram suprimidos do texto. De posse destas premissas, é possível constatar que:

³ Os testes efetuados pela EMPRESA E são caracterizados como: Alfa teste - As adaptações geradas para um determinado cliente são testadas utilizando os dados produzidos pelo ERP operante na própria fábrica. Ressalta-se que para a realização do teste em questão a fábrica de software espelha o ambiente do cliente em um laboratório de teste. Beta teste - As adaptações geradas para um determinado cliente são testadas utilizando os dados produzidos pelo ERP que opera em uma outra empresa. Para realização deste teste, a fábrica de software, solicita a uma empresa que opera com o seu ERP, dados para a realização dos testes das customizações do cliente em questão. Gama teste - Teste de aceitação, na qual as funcionalidades são testadas na presença do cliente. O teste em questão é desenvolvido dentro da atividade de implantação.

- a EMPRESA A possui em sua estrutura organizacional: Analista de Sistemas, Coordenador da Célula de Teste, Coordenador de Produção, Coordenador do Controle da Qualidade, *Software Quality Assurance* (SQA), Desenvolvedor, Diretor de Produção de Software, Equalizador, Equipe de Especificação, Gerente da Fábrica de Software, Líder de Equipe, Técnico de Configuração, Software Engineering Process Group (SEPG) e Testador;
- a EMPRESA B possui: Analista de Sistemas, Desenvolvedor, SQA, Gerente da Fábrica de Software, Gerente de Configurações, Líder de Célula e SEPG;
- a EMPRESA C possui em sua estrutura organizacional: Analista de Sistemas, Desenvolvedor, Gerente de Produção, Líder de Projeto, SEPG, SQA, Técnico em Configuração e Testador;
- a EMPRESA D possui: Área de Treinamento, Diretor, Engenheiro de Software, Gerente de Projeto, Gerente, Líder Técnico, SEPG, SQA e Técnico em Configuração;
- já a EMPRESA E possui ligados a visão processual: Diretor Geral, Equipe de Suporte, Analista de Sistemas, Grupo de Garantia da Qualidade, Programador, Programador de Produção e Técnico em Configuração;
- por fim a EMPRESA F possui: Analista de Negócio, Projetista, Célula de Teste, Gerente de Produção, Programador, Diretor de Negócio, Diretor Técnico, Líder de Célula, Técnico em Configuração, SEPG e SQA.

Por fim, um outro fato importante foi verificado na pesquisa de campo. Todas as empresas, excluindo a E, caracterizam o seu processo de produção de software como pesado, tendendo a filosofia do *Unified Process*. A EMPRESA E tende a utilizar técnicas advindas do *eXtreme Program*.

Estabelecidas as visões relacionadas aos insumos, as atividades e aos envolvidos com o processo, o protocolo utilizado para a execução do estudo de caso estabelece que se apresentem as informações relativas a gestão operacional (de projetos) e da qualidade das fábricas.

7.3 – Gestão Operacional, da Qualidade e de Configuração

Em relação à gestão de projetos, na pesquisa realizada foi constatado que as fábricas realizam as atividades de planejamento, execução (percorrendo o processo em si) e controle.

Na atividade de planejamento as fábricas recebem o projeto de software fatiado em ordens de serviços⁴ (OS) e, verificam a complexidades destas utilizando a métrica de pontos por função (exclui-se aí a EMPRESA E que consulta a sua base histórica de projeto, e de posse de uma ordem de serviço semelhante, estima o tempo de produção).

De posse da medida em pontos por função as fábricas podem estabelecer o esforço necessário para o desenvolvimento das ordens de serviços. Salienta-se que questões relacionadas a configuração do ambiente (softwares e equipamentos necessários para o desenvolvimento do projeto) e riscos, são mapeadas na atividade de planejamento do projeto.

A atividade de controle, em todas as fábricas, é realizada por meio das informações geradas com a execução do processo. Tempo que o projeto permaneceu em cada atividade, média de produtividade dos envolvidos com o projeto e o número de erros de cada ordem de serviço caracterizam algumas das medidas utilizadas na atividade em questão. As medidas citadas neste parágrafo são capturadas pela máquina de processo, detalhes teóricos sobre tal máquina foram apresentados no capítulo 5.

⁴ Uma ordem de serviço pode ser caracterizada como um conjunto de funcionalidades, pertinentes ao projeto de software, que deve ser codificada e testada.

Em todas as empresas pesquisadas a gestão da qualidade é realizada sobre duas vertentes: do produto e do processo. Na primeira, informações relativas aos números erros encontrados por produto (no caso softwares) implantados são utilizadas como medida da qualidade. A segunda utiliza o número de “desvios no processo de produção⁵” como métrica da qualidade do processo.

Por fim, questões relativas à gestão de configuração são praticadas em todas as fábricas. Para realizar tal atividade as fábricas, em um primeiro momento, controlam as versões dos artefatos do processo e dos componentes (produtos). Em um segundo momento é desenvolvido um controle de produtos disponibilizados aos clientes. A Figura A.8 apresenta a sistemática utilizada em tal questão e, nela é possível notar que um componente ou produto possui várias versões, as quais são controladas quanto a mudanças, responsável pela mudança e motivação para mesma. Na figura também é perceptível que os produtos podem ser disponibilizados para os clientes em versões diferentes, informações relacionadas a este fato também são controladas pela atividade de gestão de configuração.

7.4 – Ferramentas e Biblioteca de Componentes

Conforme apresentado no capítulo 2, uma fábrica de software deve possuir um conjunto de ferramentas para auxiliar a execução das atividades do processo de produção. As fábricas de softwares pesquisadas não fogem desta premissa e possuem, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão gerencial e processual. A relação “ferramentas, atividades e empresas” é caracterizada na Tabela 7.1⁶.

⁵ O não preenchimento de algum artefato, caracterizado como obrigatório no processo, pode ser exemplificado como desvio.

⁶ Tabela 7.1 apresenta ferramentas caracterizadas como indispensáveis na execução e controle das atividades do processo.

Empresas	Visões					
	Gerencial			Processual		
	Projeto	Configuração	Qualidade	Equalização	Codificação	Teste
EMPRESA A	1, 6	1, 2	1	1, 3, 6	1, 2, 4	1, 2, 4, 5
EMPRESA B				1, 3		1, 2, 4, 5
EMPRESA C				1		1, 2, 4, 5
EMPRESA D				1, 2, 3, 7		1, 2, 4, 5
EMPRESA E				Não pratica		1, 2, 4, 8
EMPRESA F				1, 2, 3, 6		1, 2, 4
Legenda						
[1] - Máquina de processo						
[2] - Controle de versões						
[3] - Ferramenta para visualização dos artefatos advindos da atividade de projeto de software						
[4] - Ambiente de desenvolvimento de software						
[5] - Geradores de casos de testes						
[6] - Gestão de projetos						
[7] - Ferramenta para <i>Call Conference</i>						
[8] - Editor de texto						

Tabela 7.1 – Relação das ferramentas e atividades do processo nas Fábricas de A e F. Algumas empresas citaram as ferramentas utilizadas nas atividades de modelagem de negócio, projeto e software, porém tais ferramentas não foram destacadas nesta tabela, estando descritas no anexo A, deste trabalho.

Por fim, questões relativas à formalização de uma biblioteca de componentes são praticadas apenas pelas empresas A e E.

7.5 – Produtos Gerados e Forma de Organização do Processo

Nesta seção serão apresentados os dois últimos tópicos do roteiro aplicados na pesquisa de campo. O primeiro mostra a orientação da fábrica quanto ao domínio de conhecimento, produto e tecnologia (vide Tabela 5.3). Já o segundo apresenta informações relativas à forma de organização do processo de produção das fábricas.

Em relação à orientação, é possível constatar que:

- As empresas A, C e E são orientadas a tecnologia;
- As empresas D e E são orientadas a domínios;
- A empresa E é orientada a produto.

Por fim, em relação à forma de criação e organização do processo foi possível constatar que todas as empresas pesquisadas instituíram seu processo a partir das experiências dos profissionais. Além de tal experiência, as empresas utilizaram o modelo CMMI como guia de melhoria de processo. A palavra guia é bem empregada, pois aspectos como definição de domínios de aplicações e modelagem tecnológica relacionada à máquina do processo, que são de fundamental importância em um ambiente fabril de produção de software, não são contemplados em tal modelo.

Apresentada a estruturação do processo de produção das empresas brasileiras, o próximo capítulo apresenta uma análise comparativa entre as fábricas brasileiras e estrangeiras.

Capítulo 8 – Análise Comparativa entre as Fábricas Brasileiras e Estrangeiras

Este capítulo tem como objetivo apresentar a relação teórica entre as fábricas brasileiras e estrangeiras. A descrição do processo de produção de tais empresas encontram-se condensadas na seção 2.1 e no capítulo 7 e detalhadas nos anexos A e B. Salienta-se que a relação teórica será concebida segundo o referencial bibliográfico apresentado no capítulo 2 e no modelo fabril proposto pelo ELABSOFT, delineado no capítulo 3.

No referencial teórico sobre fábrica de software, é possível encontrar os atributos inerentes a uma estrutura fabril de produção de software (vide Quadro 2.1). De posse de tais atributos e dos casos (empresas brasileiras e estrangeiras) é possível perceber que:

1. Todas as empresas pesquisadas possuem um processo de produção de software definido e institucionalizado.
2. Em relação ao alto poder de atendimento, não se pode afirmar nada em relação às empresas, pois dados quantitativos sobre o número de clientes atendidos, diversidade de aquisição de produtos e tempo de espera de atendimento não foram divulgados nas entrevistas e não estão explicitados na obra de CUSUMANO (1991).
3. Todas as empresas brasileiras trabalham com o conceito de ordens de serviços (vide seção 7.3), destaque para:
 - empresas A, C, D e E: as ordens são geradas na atividade de projeto de software (vide anexo A);
 - EMPRESA B: as ordens são geradas na atividade de equalização.

Já nas fábricas estrangeiras, o conceito de ordem de serviço aparece, explicitamente, somente no caso da NEC. Porém ao analisar as seções do anexo B que tratam as questões sobre “processo de produção”, “gestão de projetos” e “teste de software”, é perceptível que as demais empresas estrangeiras, dividem a produção do software em pequenas partes (componentes), distribuindo-as sobre as atividades do processo. A SDC possui em seu modelo de organização de produção, o conceito de células de produção ou equipes de produção (vide Figura A.1). A Hitachi divide o projeto em módulos, estes, por sua vez, são divididos em programas. Na Toshiba as unidades de trabalhos são responsáveis pela execução de um conjunto de tarefas. A Fujitsu divide o projeto de software em um conjunto especificações. Todas estas divisões se assemelham a idéia de divisão do projeto de software em OS praticada pelas empresas brasileiras e pela NEC.

4. A estimativa de custo e prazo¹ é uma constante nas empresas apresentadas neste trabalho. Tais estimativas são armazenadas, em bases históricas de projetos.
5. O controle de perfis de recursos humanos foi detectado, explicitamente, nas empresas brasileiras. Todas elas possuem funções bem definidas e aglutinadas nas visões gerencial e processual, a seção 7.2 e as Figuras A.6, A.13, A.16, A.22, A.26 e A.30 confirmam esta afirmação. Já na obra de CUSUMANO (1991) é possível encontrar seções relacionadas ao plano de carreira praticado pelas empresas, porém estas não provêem informações suficientes para classificar os perfis profissionais dentro das visões citadas.
6. A atividade de planejamento e controle da produção é desenvolvida em todas as fábricas apresentadas. Ressalta-se que as empresas japonesas possuem maior rigidez, em relação a tal controle, as tabelas B.3, B.4, B.5 e B.9, provêem margem para que autor deste trabalho faça esta afirmação.

¹ HUMPHREY (2001) refere-se a tal item como estabelecimento de métricas produtivas ligadas ao processo de produção.

7. O controle de componentes é uma prática em todas as empresas estrangeiras pesquisadas. Dentre as empresa brasileiras, somente a A e a E produzem software utilizando tal prática (vide seção 7.4).
8. Todas as empresas utilizam métodos e técnicas padronizadas na produção do software. A própria definição e institucionalização do processo induz a isto. Este fato, também, é perceptível ao analisar os dados espelhados nas tabelas B.3, B.4, B.5 e B.9 (casos nipônicos). Seria impossível capturar tais dados sem a utilização de métodos e técnicas padronizadas. A SDC, também, atende a estas características, visto que uma das práticas abordadas pela empresa é promover o conceito de “repetibilidade”² no processo de desenvolvimento de software.
9. A garantia da qualidade do produto é praticada pelas empresas dentro da atividade de teste. Nas empresas brasileiras, ressalta-se que a máquina de processo provê subsídios estatísticos para a verificação de tal atributo. O campo “erros encontrados a partir da atividade de teste”, apresentado na Tabela A.2 provê subsídios para a confirmação de tal fato. Neste item é importante ressaltar a fábrica de software da Toshiba implementa o conceito de “rastreabilidade” e captura de um erro antes que ele atinja a atividade de teste.
10. A melhoria contínua do processo é uma constante nas empresas que se norteiam pelo modelo CMMI, cinco dos seis casos brasileiros. Este atributo, também, é verificável nos casos estrangeiros (vide seção 2.1.3). Já a EMPRESA E, certificada com ISO 9001, também, está preocupada com os aspectos relacionados à melhoria contínua do processo, pois em sua estrutura organizacional é contemplado o grupo de garantia da qualidade (vide seção 7.2).
11. Por fim, foi constatado que as empresas brasileiras possuem uma boa infraestrutura de hardware para o desenvolvimento de software (fato este comprovado, pois o autor deste trabalho as visitou). Porém este fato não pode

² A “repetibilidade” só ocorre após a padronização.

ser estendido para as empresas estrangeiras, pois a coleta de dados para estes casos foi desenvolvida por meio do referencial bibliográfico (CUSUMANO (1991)). Em relação à automação do processo, foi verificado que todas as empresas, tanto as brasileiras quanto as estrangeiras, possuem um conjunto mínimo de ferramentas que auxiliam os envolvidos com o processo na execução de suas atividades (vide Tabela 7.1). Destaque para a EMPRESA D e para a Hitachi que configuraram suas ferramentas relacionadas a visão gerencial de forma integrada às ferramentas situadas na visão processual.

Além dos atributos apresentados no quadro 2.1, o capítulo 2, apresenta por meio da Figura 2.1 um modelo de fabril orientado a qualidade (proposta de CANTONE (1992)). Tal modelo é, totalmente, aderente a processo de produção das empresas. Tal fato pode ser comprovado ao analisar os seguintes aspectos:

1. *SF MODELS* ou modelo de produção (caracterizados nos ciclos longo, médio e curto). Todas as fábricas, apresentadas nos capítulos 2 e 7, foram caracterizadas em relação a tais ciclos;
2. O modelo de processo utilizado pelas fábricas brasileiras possui característica incremental, excluindo a EMPRESA F que utiliza como modelo, um misto de prototipação e incremental. Este aspecto não pode ser generalizado para as fábricas estrangeiras, pois não foram encontradas informações suficientes, no referencial bibliográfico, que pudessem classificar tais fábricas nos modelos de processo apresentados no capítulo 4.
3. Os modelos formais da qualidade de software (como o CMMI) orientam 5 das 6 fábricas nacionais (exclusão apenas da EMPRESA E). As empresas estrangeiras não utilizam modelos formais relacionados à qualidade, porém é perceptível que os aspectos relacionados à qualidade do produto e do processo são, totalmente, pertinentes a elas.

Além dos atributos e do modelo proposto por CANTONE (1992), LI et. al. (2001) configuram uma estrutura genérica para a definição do conceito de fábrica (vide Figura 2.2). As empresas brasileiras e estrangeiras aderem quase que, totalmente, a tal estrutura.

Nos casos brasileiros, foi possível verificar que o suporte técnico ao processo é suprido, na maioria das vezes, pelos componentes de tecnologia (classificados, no capítulo 6, como ferramentas) – aderência a entidade 1 da Figura 2.2. O processo é dividido em atividades e tarefas e regido pelo modelo da qualidade CMMI, em 5 dos 6 casos – aderência a entidade 2 (Figura 2.2). Os envolvidos com o processo, também, foram mapeados, formalmente, porém em nenhuma empresa existem práticas relacionadas ao TSP e ao PSP (fato este que levou a utilização do termo “quase que totalmente” no parágrafo anterior) – aderência a entidade 3. Questões sobre gerenciamento da qualidade, organização do modelo do processo e estrutura organizacional da fábrica de software, também, foram detectadas em todos os casos (veja relação dos casos com o modelo de CANTONE, anteriormente, apresentada) – aderência a entidade 4. Ressalta-se, também, que algumas empresas possuem certificação da qualidade ISO 9001, como é o caso das empresas A e E aderência a entidade 5. Por fim, o armazenamento de dados é configurado sobre o conceito base histórica de projetos – aderência a entidade 5 da Figura 2.2.

Já as empresas estrangeiras aderem, plenamente, à entidade técnicas (entidade 1 da Figura 2.2), pois é visível em todas elas a utilização das ferramentas e do conceito de reuso de componentes. O processo de produção em tais empresas não possui certificação alguma (análise efetuado com base nos relatos de CUSUMANO (1991)) e as mesmas considerações delineadas para as empresas brasileiras sobre as entidades 3, 4 e 5 são válidas para os casos estrangeiros.

Uma outra vertente conceitual sobre os modelos fabris, apresentados neste trabalho, é proposta por BASILI et. al. (1992) (vide Figura 2.5). A aderência das empresas brasileiras e das estrangeiras ao modelo segmentado, proposto pelo autor, é caracterizada no contexto da organização do trabalho, visto que todas

possuem potencial para o desenvolvimento de software baseado em componentes, porém algumas não estabelecem um processo formal ligado a isto (caso das empresas B, C, D e E).

Já em relação à estruturação física, somente a fábrica de software da EMPRESA A possui uma unidade de produção de componentes operando de forma separada. Na EMPRESA E, a técnica de reuso é concebida durante a execução das atividades do processo, por exemplo: no momento da codificação de algum ativo de software a equipe verifica a possibilidade de se estruturar a idéia de reuso, e o faz.

No âmbito internacional é possível destacar duas empresas que praticam um alto grau de reuso, a Toshiba e a Fujitsu. Já em relação à estruturação física de tais fábricas não se pode afirmar nada, pois o autor deste trabalho não as visitou para o desenvolvimento deste estudo.

Uma relação teórica entre os casos e o modelo de fábrica de programas, proposto por FERNANDES e TEIXEIRA (2004) e apresentado no capítulo 2, por meio da Figura 2.8, também, é estabelecida neste capítulo³. Em sua proposta os autores estabelecem que uma fábrica de programas deve possuir as seguintes atividades:

- Recebimento e aceitação de artefatos: Alusão à atividade de equalização presente em todos os casos brasileiros. Já nos relatos apresentados por CUSUMANO (1991) tal atividade não é estabelecida, formalmente;
- Planejamento e distribuição (das ordens de serviços): As atividades de planejamento e distribuição, também, são contempladas nos casos apresentados nos capítulos 2 e 7 e nos anexos A e B. Para os casos brasileiros este fato é perceptível nas figuras A.7, A.14, A.17, A.19 e nos relatos apresentado na seção 7.3. Para as empresas estrangeiras, exemplos deste fato são explicitados no

³ A relação ocorre sobre a ótica do ciclo curto, presente em todas as fábricas delineadas neste trabalho.

caso da Hitachi, Toshiba e Fujitsu. A Hitachi divide os projetos em módulos e os módulos, por sua vez, são divididos em programas, os programas são desenvolvidos por um conjunto de atores pertinentes ao processo (aderência ao conceito de distribuições proposto por FERNANDES e TEIXEIRA (2004)). Por fim, Toshiba e a Fujitsu desenvolvem um controle de produtividade norteado pela seguinte métrica: linhas de código produzidas por programador, fato este que nos leva a crer que as empresas em questão, também, praticam a idéia de planejamento e distribuição de suas ordens de serviço;

- Análise de tarefa⁴: Esta atividade não foi detectada nas fábricas de software pesquisadas;
- Codificação, Teste e Homologação: Atividades presente em todos os casos descritos. Em algumas empresas a homologação é chamada de entrega, como é o caso da SDC e da NEC. Em outras, como na Toshiba, a Homologação é, simplesmente, tratada como teste de software. A Fujitsu, por sua vez, trata a questão da homologação como teste de sistema. As empresas B, C e D, utilizam o vocábulo teste de integração para designar tal termo. Por fim, a empresa E utiliza os termos alfa e beta teste para designar a atividade de homologação.

FERNANDES e TEIXEIRA (2004), também, classificam uma organização fabril para o desenvolvimento de software em dedicada e não dedicada. Analisando as considerações efetuadas pelos autores e com base nos relatos desenvolvidos no capítulo 7 e no anexo A é possível verificar que:

- A EMPRESA A é especializada nas plataformas JAVA e .NET; as células de produção são organizadas por plataforma tecnológica (característica não dedicada) e por cliente (característica dedicada). Em relação aos aspectos contratuais, a empresa estabelece, para a maioria dos projetos, procedimentos para a contratação interna da fábrica de programas para o desenvolvimento do

⁴ Nesta atividade, a ordem de serviço é padronizada seguindo os padrões pré-estabelecidos para codificação.

produto (característica não dedicada). Aspectos de gestão de projetos possuem configuração local, isto é, a fábrica de software, caracterizada como ciclo curto é, relativamente, independente para gerenciar os seus projetos, tendo que atender somente as prerrogativas estabelecidas nos aspectos de contratação⁵ (característica dedicada);

- A EMPRESA B aceita qualquer tipo de projeto em qualquer plataforma, criando segmentos relacionados ao domínio da aplicação (característica dedicada). Ressalta-se que a empresa não é orientada a domínio, uma segmentação é criada toda vez que um projeto pertinente a um domínio não desenvolvido pela empresa é concebido. Organiza as células de produção por plataformas (característica não-dedicada). Diferentemente da EMPRESA A, a fábrica de projetos sempre contrata a fábrica de programas (ciclo curto) (característica dedicada);
- A EMPRESA C, também, aceita qualquer tipo de projeto nas plataformas JAVA .NET, criando segmentos relacionados ao domínio da aplicação. (característica dedicada). As células de produção são organizadas por plataforma e, posteriormente, por domínios da aplicação (característica não-dedicada). Tem como regra a contratação da fábrica de programas (caracterizada como ciclo curto) pela fábrica de projetos (característica dedicada);
- A EMPRESA D aceita projeto em qualquer plataforma, criando segmentos relacionados ao domínio da aplicação (característica não-dedicada). As células de produção são organizadas por plataforma (característica não-dedicada). A contratação interna da fábrica de programas não é uma prática obrigatória na empresa (característica não-dedicada);
- A EMPRESA E aceita projetos em uma plataforma e atua somente em um segmento (característica dedicada). Organiza suas células de produção por

⁵ Esta informação, também, é válida para as demais fábrica brasileiras.

cliente (característica dedicada). A organização baseada em projetos sempre utiliza os serviços da fábrica de código (característica dedicada).

- A EMPRESA F aceita projetos em qualquer plataforma e em qualquer segmento, desde que este possua um alto grau de replicabilidade do produto (característica não-dedicada). Organiza as células de produção por cliente (característica dedicada). A fábrica de software de ciclo curto sempre é contratada para o desenvolvimento do produto (característica dedicada).

A relação entre o modelo proposto pelo ELABSOFT, seção 3.1, e as fábricas, apresentadas nos capítulos 2 e 7, também, foi mapeada neste trabalho.

O modelo proposto pelos pesquisadores do ELABSOFT está, organizado, em 4 visões: metodológica, tecnológica, negócio e fabricação de componentes. Olhando sob a ótica da organização do trabalho, todas as empresas pesquisadas podem ser inseridas dentro das visões, nominalmente, citadas. Porém, ao se analisar a estrutura física das fábricas, é perceptível que a visão de fabricação de componentes é incorporada pela visão do negócio. Nas fábricas brasileiras, exceto a EMPRESA A, não foram detectadas unidades exclusivas com a responsabilidade de desenvolver, somente, componente de modelagem ou de software. Isto não quer dizer que tais empresas não praticam o reuso, mesmo que informalmente. Nestes casos, os componentes são desenvolvidos dentro das próprias atividades do processo. De posse desta informação é possível verificar que a estrutura física das fábricas (a maioria delas) está configurada de acordo com a Figura 8.1.

Figura 8.1 – Estruturação física das fábricas brasileiras (figura derivada do processo fabril apresentado na Figura 3.6).

Além das quatro visões, os autores do modelo utilizado pelo ELABSOFT, deixam claro que um processo de produção de software deve possuir atividades relacionadas à: análise de sistemas, projeto de software, construção do software, teste, armazenamento e revisão. As atividades do processo de produção das fábricas brasileiras e estrangeiras seguem a mesma configuração. A averiguação deste fato pode ser constatada na estrutura do processo de produção de tais empresas apresentada nos capítulos 2 e 7 e nos anexos A e B. A Tabela 8.1 apresenta a aderência das atividades do processo de produção do software das empresas ao modelo do ELABSOFT.

	BRASILEIRAS (EMPRESAS)						AMERICANA	NIPÔNICAS			
ELABSOFT	A	B	C	D	E	F	SDC	HITACHI	TOSHIBA	NEC	FUJTSU
Contratação	P	P	P	P	P	P	P	P	P	P	P
Analisar	P	P	P		P	P	P	P	P		P
Projetar	P	P	P	P	P	P	P	P	P	P	P
Construir	P	P	P	P	P	P	P	P	P	P	P
Integrar	P	P	P	P	P ⁶	P	P	P	P	P	P
Implantar	P	P	P		P	P	P	P	P	P	P
Revisar					P						
Gestão Projetos	P	P	P	P	P	P	P	P	P	P	P
Manutenção ⁷	P	P	P	P	P	P	P	P	P	P	P

Tabela 8.1 – Presença (P) das atividades do processo de produção de software do modelo ELABSOFT nas empresas pesquisadas

Ao analisar a Tabela 8.1 é possível perceber que todas as atividades propostas no modelo do ELABSOFT são encontradas, em pelo menos uma das empresas. Porém, todas as atividades encontradas, explicitamente, nas empresas não são presenciadas no modelo, destaque para: o de levantamento de requisitos, a equalização, a gestão de configuração e da qualidade do produto e do processo. De posse deste fato, o autor deste trabalho, propõe, por meio da Figura 8.2, uma nova configuração para o modelo em questão. Nesta proposta, é perceptível, destacado em vermelho, a:

⁶ Utiliza o conceito de teste de imersão: dividido em alfa, beta e gama teste (vide seção A.5.3)

⁷ A atividade de manutenção no modelo proposto pelo ELABSOFT está ligada a solicitação de melhoria (vide figura 3.6). Tal solicitação dispara as atividades de projeto, construção, teste e implantação. Este fato também pode ser percebido, informalmente, nas fábricas pesquisadas. Geralmente, a atividade manutenção, nas fábricas pesquisadas, gera uma ordem de serviço e esta percorre todas as demais atividades do processo.

- Gestão da qualidade do produto, ocorrendo, paralelamente, a atividade de teste e após o atendimento das necessidades mercadológicas (alusão à gestão da qualidade do produto, praticadas pelas fábricas de software, principalmente, as nipônicas);
- Equalização do modelo de negócio (presente no caso da EMPRESA D) e do projeto de software;
- Gestão de configuração: Ocorrendo sobre a base de componentes e após o atendimento das necessidades mercadológicas.

Finalizando a análise comparativa entre as fábricas, apresentadas nos capítulos 2 e 7 e nos anexos A e B, e o modelo de processo desenvolvido pelos pesquisadores do ELABSOFT, é possível notar que a “maioria” dos papéis ativos em uma estrutura fabril possui uma relação com o envolvidos no processo de produção de software das fábricas pesquisadas (este fato, também, é destacado por FABRI et. al. (2006b)). Se uma análise minuciosa for desenvolvida, é válido afirmar que as funções e responsabilidades dos envolvidos com o processo de produção estão distribuídas em tais papéis, como por exemplo: em algumas fábricas, é possível verificar a presença do técnico em configuração (empresas A, C, D, E e F), no modelo proposto pelo ELABSOFT é verificável que tal função é desempenhada pelo gerente de configuração (mesmo termo adotado pela empresa B). O termo “maioria” foi destacado, propositalmente, pois no modelo contemplado, pelo ELABSOFT, não foram inseridas as figuras do SQA, da SEPG e do Equalizador, papéis estes, presente em quase todas as fábricas (brasileiras) pesquisadas. De posse de tal constatação, ou autor deste trabalho, concluí que tais papéis devam ser incluídos no modelo proposto pelo ELABSOFT.

Por fim, uma relação consolidada entre as empresas brasileiras, japonesas e americana, baseada no roteiro para desenvolvimento do estudo de caso, apresentado na tabela 6.3, pode ser verificada na Tabela 8.2.

	Empresas Brasileiras						Empresas Nipônicas e Americana				
	EMP. A	EMP. B	EMP. C	EMP. D	EMP. E	EMP. F	SDC	Hitachi	Toshiba	NEC	Fujitsu
Insumos ⁸	Longo	Longo	Longo	Médio+	Longo	Longo	Longo	Longo	Longo	Médio	Longo
-Caracterização	Pesado	Pesado	Pesado	Pesado	Técnicas <i>xP</i>	Pesado	Pesado	Pesado	Pesado	Pesado	Pesado
Gestão	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza	Realiza
Ferramentas**	MP. GP. CV.	MP. GP. CV. GT.	MP. GP. CV. GT.	MP. CC. GT. CV. GP. ***	MP. GP. CV.	MP. GP. CV.	MP. GP.	CASE. CAPS. ****	GR. GP. MP*****.	Prototipagem GC GT. GP. CV.	GC. GD. Gestão de Teste. GP.
Reuso	Formal	Informal	Informal	Informal	Formal	Informal	Formal	Formal	Formal	Formal	Formal
Domínio*	Não	Não	Não	Sim	Sim	Sim	Não	Não	Não	Não	Não
Produto*	Não	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Não
Tecnologia*	Sim	Não	Sim	Não	Sim	Não	Não	Não	Não	Não	Não
Criação Processo	Não utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Utilizou estrutura formal	Não utilizou estrutura formal	Não utilizou estrutura formal	Utilizou estrutura formal

Tabela 8.2 – Relação entre as empresas brasileiras e estrangeiras (base: roteiro para o desenvolvimento do estudo de caso – Tabela 6.3). Legenda: Longo = Ciclo Longo de Produção de Software.

As empresas caracterizam unidades organizacionais, denominam-nas como fábricas, e estas por sua vez praticam somente o ciclo curto de produção (equalização, codificação e teste). +Analisando somente a unidade visitada. MP = Máquina de processo. GP = Gestão de Projetos. CV = Controle de Versões. GT = Geradores de caso de teste. ** Todas as fábricas utilizam ferramentas de ligadas ao ambiente de desenvolvimento (por exemplo: editores gráficos para o desenvolvimento de diagramas e compiladores). CC = Ferramenta de *Call Conference*. ***Ambiente Integrado. CAPS = *Computer-Aided Software Production System* – Integrada com a ferramenta CASE. ND = Não detectada. GR = Gestão de Requisitos. ****Ferramentas integradas: aglutinam as funcionalidades da máquina de processos. *****Denominada pela empresa como controle de produtividade. GC = Gerador de Código. GD = Gestão de documentos. -Caracterização = Leve ou Pesado. *Forma de orientação das empresas.

⁸ As empresas brasileiras se caracterizam com o ciclo longo, porém o princípio de produtividade em alta escala, com tarefas bem definidas, métricas estabelecidas, e um alto poder de controle da produção são classificadas dentro da ótica das unidades organizacionais que praticam o ciclo curto (vide seção 7.2).

Capítulo 9 – A Aderência do Modelo Proposto neste Trabalho aos Casos Brasileiros e Estrangeiros

Este capítulo tem como objetivo verificar a aderência dos casos ao modelo, proposto preliminarmente, no capítulo 5. Ressalta-se que ao realizar tal verificação, a proposta pode evoluir ou sofrer alterações.

Para estruturar este capítulo, a representação gráfica do modelo, mapeada na Figura 5.5, será utilizada como um guia na concepção das seções. É importante ressaltar que às atividades de simulação, execução, avaliação e armazenamento serão verificadas no capítulo 10¹.

9.1 A Aderência dos Casos aos Requisitos

Esta seção tem como objetivo verificar a aderências dos requisitos do modelo proposto, preliminarmente, no capítulo 5 (domínio do conhecimento, modelo da qualidade, tecnologia, ciclo de produção e atores) aos processos fabris apresentados nos capítulos 2 e 7 e nos anexos A e B.

O primeiro requisito que será verificado é a definição de domínio do conhecimento por parte das empresas. Ao analisar os relatos apresentados nos capítulos 2 e 7 é possível constatar que as empresas A, B, C, E e as estrangeiras não trabalham sobre a orientação em questão. Já as empresas D e F preservam tal característica. Este fato nos leva a crer que antes de definir o domínio do conhecimento, é necessário definir qual estratégia de mercado que a fábrica de software irá adotar. Produzir para um segmento apenas, ou estar aberta aos vários modelos de negócios adotados no mercado? A crença ressaltada promove uma

¹ As atividades de seleção e adaptação não serão verificadas neste trabalho, pois somente um processo de produção de software foi criado com o modelo proposto.

alteração no modelo. Se, estrategicamente, a empresa optar produzir para um segmento apenas, é que este segmento deve ser definido.

Já em relação aos modelos da qualidade é perceptível que cinco das seis empresas brasileiras, pesquisadas possuem certificação CMMI. As empresas estrangeiras não são certificadas em um modelo, porém todas elas preocupam-se com tal fato (vide as considerações, sobre a questão da qualidade, apresentadas no capítulo 8).

A orientação tecnológica é uma constante nas empresas A, C e E. As demais não são regidas sob a luz deste fato, porém estas organizam sua produção em células, que na maioria das vezes são orientadas por segmento de produto, por cliente e, conseqüentemente, por tecnologia. Com isto é válido dizer que a orientação tecnológica ocorre no contexto do projeto e esta deve ser contemplada no modelo proposto.

Em relação aos ciclos de produção, foi verificado que as empresas de software analisadas trabalham sob a luz do ciclo longo (exclui-se a EMPRESA D e a NEC, que praticam o ciclo médio). Já as fábricas são caracterizadas como unidades produtivas responsáveis pela equalização, codificação e teste. Com isto, elas são regidas pelo ciclo curto. Estes fatos promovem a aderência do ciclo de produção, delineado no modelo, com as empresas.

Por fim, ao analisar o processo de produção das fábricas brasileiras é perceptível a presença dos seguintes atores (vide capítulo 7):

- Programador ou desenvolvedor: Presente nas empresas A, B, C, E e F. Na EMPRESA D esta atividade é praticada pelo engenheiro de software;
- Equalizador: Presente na EMPRESA A nas demais este papel é desenvolvido pelo programador ou pelo desenvolvedor;

- Líder de equipe, líder de projeto, líder técnico ou líder de célula: Presente em todas as empresas, exceto na E. Nela, tal papel é praticado pelo programador da produção;
- Técnico em configuração: Presente em todas as empresas, exceto na B. Nesta empresa tal papel é praticado pelo gerente de configuração;
- Analista de Sistemas: Presente nas empresas A, B, C, E e F. Na EMPRESA D este papel é desempenhado pelo engenheiro de software;
- Testador ou célula de teste: Papel presente nas empresas A e C. Nas demais, tal tarefa é desempenhada pelo desenvolvedor (empresas B, E e F) e pelo engenheiro de software na EMPRESA D;
- SEPG e SQA. Todas as empresas, exceto a E², possuem este papel definido.

De posse dos itens acima é interessante que visão processual (vide Figura 2.3) de uma fábrica de software possua o seguinte corpo de profissionais: programadores, analistas de sistemas, equalizadores, testadores, líder de célula, técnico em configuração, gerente de projeto (este assumindo o papel do planejamento e programação da produção), SEPG e SQA.

9.2 A Aderência dos Casos a Atividade de Modelagem

Conforme relatado no capítulo 5, a modelagem do processo de produção de software deve ser elaborada utilizando uma PML. O modelo preliminar prevê a utilização do atigrama definido pela normalização IDEF-0. Para verificar se a adaptação adere aos casos, será modelado, a título ilustrativo, o processo de

² Na EMPRESA E a função da SEPG é desempenhada pelo grupo de garantia da qualidade. Já a função de SQA é desenvolvida pelo programador de produção.

produção de software da EMPRESA A³. A modelagem dos demais casos segue a mesma linha, motivo este que isenta a configuração de tal atividade para todos os casos e todas as atividades.

A Figura 9.1 e a Figura 9.2 apresentam a modelagem do processo de produção de software da EMPRESA A de forma, estratificada, isto é: o processo não está detalhando. Efetuando uma análise mais aprofundada nas figuras é perceptível que o processo pode ser verificado segundo as perspectivas (características pertinente à atividade de instanciação) de:

- Fluxo de trabalho: Basta indexar a leitura da figura com base nas atividades, representadas pelos retângulos;
- Fluxo de dados: Basta indexar a leitura da figura por meio dos fluxos de saída (ERS, MS, PSW/OS, entre outros);
- Recursos: Basta indexar a leitura da figura por meio das ferramentas;
- Habilidades: Basta indexar a leitura da figura por meio dos fluxos que representam os envolvidos com as atividades.

³ O fato que motivou à escolha da EMPRESA A para compor esta seção, está ligado ao tempo e a quantidade de pessoas entrevistadas. Na visita efetuada a tal empresa foram gravadas 20 horas de entrevistas, e o número de entrevistados foram 5 (vide Tabela 6.2).

Figura 9.1 – Modelagem do processo de produção de software da EMPRESA A. Nota: Parte das legendas estão descritas na Tabela 9.1. Nesta mesma tabela também é possível encontrar a descrição dos objetivos das atividades

Figura 9.2 – Modelagem das atividades de gestão de projeto, configuração e da qualidade

RE-4	Verificar a "corretude" e a "completude" das especificações.
FE-4	Máquina de processo e ferramenta para controle de versões.
HA-4	A atividade requer habilidades equalizadoras. Participam desta atividade: <u>Equalizador</u> (principal envolvido), Desenvolvedor, Equipe de Especificação, Líder de Equipe, Técnico em Configuração, SQA.
RE-5	Verificar se os artefatos da OS foram equalizados. Verificar se os artefatos seguem um padrão pré-definido.
FE-5	Ambiente de desenvolvimento para JAVA e .NET, controle de versões e máquina de processos.
HA-5	A atividade requer habilidades de desenvolvimento de software nas tecnologias JAVA e .NET. Participam desta atividade o: Desenvolvedor, Equalizador, SQA e Líder de Equipe.
RE-6	Respeitar os <i>checklist</i> para cada tipo de componente a ser testado.
FE-6	Máquina de Processo. Geradores de casos de testes. Controle de versões.
HA-6	A atividade requer habilidades e conhecimentos em técnicas de teste. Participam desta atividade o: Testador, Desenvolvedor, Líder de Equipe, Técnico em Configuração e SQA.
Descritor das atividades de:	
<p>EQUALIZAÇÃO: A atividade de equalização tem como objetivo verificar a "corretude" e a "completude" das especificações advindas da atividade de projeto de software para que as mesmas possam ser colocadas em linha de produção (codificação e teste).</p> <p>CODIFICAÇÃO: Codificar as ordens de serviços advindas da atividade de equalização.</p> <p>TESTE: A atividade de teste tem como objetivo efetuar o teste de caixa preta nas ordens de serviço.</p> <p>GESTÃO DE PROJETOS: Atividade é focada no gerenciamento de custo, cronograma, recursos e riscos relacionados ao desenvolvimento de software.</p> <p>GESTÃO DE CONFIGURAÇÃO: Em um primeiro momento é feito um controle de versões de artefatos do processo e de componentes (ou produtos). Em um segundo momento é desenvolvido um controle de produtos disponibilizados aos clientes.</p> <p>GESTÃO DA QUALIDADE: A gestão da qualidade é desenvolvida a partir das informações geradas pela máquina de processo, principalmente, no que se diz respeito ao controle de erros mapeado na atividade de teste. Além destas informações, os desvios no processo, também, servem de parâmetro da qualidade. Reuniões periódicas são realizadas com o objetivo de apresentar o desempenho da equipe de produção da fábrica de software e propor metas da qualidade a serem perseguidas.</p>	

Tabela 9.1 – Legendas e descritores da Figura 9.1.

Além das perspectivas, é verificável na Figura 9.1 e na Figura 9.2 as operações de composição e herança. A composição está relacionada às tarefas e atividades anteriores e posteriores se comparadas à tarefa corrente. Por exemplo, a atividade de codificação, quando unida às atividades de equalização e teste compõem o conceito de ciclo curto de produção. Por fim, a atividade de codificação foi gerada a partir da instanciação deste mesmo ciclo e herda, na sua composição processual, as informações geradas nas atividades de levantamento de requisitos, análise e projeto de software (atividades, estas, inerentes ao ciclo médio e longo).

9.3 A Aderência dos Casos à Atividade de Instanciação do Modelo

A atividade de instanciação parte das informações geradas na atividade de modelagem e concebe as tarefas inerentes ao processo. Além das tarefas, os ativos do processo são configurados, os recursos humanos e tecnológicos (informações estas advindas da atividade de requisitos do modelo) são alocados, a máquina de processo é delineada e o cronograma para implantação do processo é definido.

É importante ressaltar nesta seção que, a título ilustrativo, somente as tarefas da atividade de equalização serão instanciadas, as demais seguem o mesmo princípio. A alocação dos recursos humanos e tecnológicos que foi verificada, previamente, na atividade de modelagem, será refinada e por fim, a máquina de processo será delineada. As informações relacionadas ao cronograma de implantação do processo não serão apresentadas, visto que o processo já se encontra institucionalizado em tal empresa.

Com base nos relatos apresentados na seção A.1.3, a atividade equalização da EMPRESA A possui as seguintes tarefas:

- Tarefa um (T1) – Verificação:
 - Objetivo: verificar a “corretude” e a “completude” das ordens de serviços.

- Tarefa dois (T2) – Contatar equipe de projeto de software:
 - Objetivo: Se a ordem de serviço estiver inconsistente e incompleta a pessoa responsável entra em contato com a equipe de projeto de software para sanar as possíveis dúvidas. Importante: se as dúvidas não forem sanadas, a equalização pode ser rejeitada (para rejeitar a equalização, o equalizador utiliza um guia de pontuação). Neste caso o documento de rejeite deve ser preenchido.
- Tarefa três (T3) – Disparar a atividade de codificação:
 - Objetivo: Se a ordem serviço estiver consistente, completa e correta a atividade de codificação é disparada.
- Tarefa quatro (T4) – Preenchimento do relatório da equalização:
 - Objetivo: A equalização demanda o preenchimento de um relatório, em tal documento são descritos os problemas e o histórico relacionado à equalização de uma ordem de serviço.
- Tarefa cinco (T5) – Apontar as atividades realizadas na máquina de processo.
 - Objetivo: Neste apontamento o equalizador deve informar as horas trabalhadas em relação às ordens de serviços.

De posse da descrição das tarefas, é possível modelá-las utilizando a PML apresentada no capítulo 4 (vide Figura 9.3).

A Figura 9.3 configura-se como um detalhamento da atividade de equalização (vide Figura 9.1), e apresenta as tarefas descritas anteriormente, as habilidades necessárias para execução delas, os recursos tecnológicos necessários para o desenvolvimento de cada uma e, por fim, os fluxos de dados que interligam cada

atividade. Ressalta-se que as mesmas considerações relacionadas às perspectivas de fluxo de trabalho, fluxo de dados, recursos e habilidades, também, são válidas para o detalhamento da atividade em questão.

Figura 9.3 – Modelagem das tarefas da atividade de equalização

De posse da Figura 9.3 é possível perceber, também, a presença dos seguintes ativos do processo (ou artefatos): Guia de pontuação, *checklist* para verificação, documento de rejeite e relatório de equalização. Para fins ilustrativos, será apresentado, na atividade de instanciação, o documento de rejeite⁴. A apresentação do documento de rejeite segue as considerações apresentadas na base de moldes, definida no capítulo 5 (vide Quadro 9.1).

⁴ Ressalta-se que autor deste trabalho não obteve uma autorização formal da empresa em questão para divulgar todos os artefatos do processo

Quadro 9.1 – Apresentação do artefato: Documento de rejeite – EMPRESA A

<u>EMPRESA A</u>		
Projeto: <<Inserir o Nome do Projeto>>		
DOCUMENTO DE REJEITE		
Versões:		
<u>Responsável</u>	<u>Data de Alteração</u>	<u>Histórico de Revisão no Documento</u>
SQA e SEPG	20/10/2006	Criação do documento
Responsável autorizado pelo preenchimento do ativo: Equalizador		
Introdução:		
<p>Este documento tem como finalidade apresentar o motivo pelo qual uma ordem de serviço foi rejeitada. O artefato em questão influencia, diretamente, toda a atividade de equalização. Quando tal documento é preenchido, a atividade de projeto de software deve especificar, novamente, os artefatos que descrevam as funcionalidades que compõem a ordem de serviço em questão. Este artefato relaciona-se, diretamente, com a ordem de serviço e todos os artefatos anexos a ela, o guia de pontuação, o <i>checklist</i> para verificação e o relatório de equalização.</p>		
Descrição de Rejeite da OS		
OS número:	Motivo da rejeição:	Sugestões para melhoria:

Por fim, a máquina de processo da EMPRESA A descrita na seção A.1.5 pode ser relacionada à formalização direcionada pelas tuplas 1, 2, 3, 4, 5 e 6 apresentadas no capítulo 5. Ao analisar a estrutura da máquina é possível perceber que as funcionalidades:

1. Manter os clientes e seus projetos de software: Funcionalidade esta que se relaciona com as variáveis i e p da tupla G (tupla 3);
2. Manter os produtos desenvolvidos, produtos estes relacionados às ordens de serviços: Funcionalidade esta que se relaciona com a tupla C (tupla 1);
3. Armazenar os envolvidos, erros e o tempo de desenvolvimento para cada produto: Funcionalidade esta que se relaciona com as variáveis v da tupla N e a da tupla E ;

4. Relacionar os produtos desenvolvidos aos projetos dos clientes: Relação esta espelhada nas variáveis p e C da tupla G .

Para consolidar a relação desenvolvida, a Tabela A.2 que apresenta as informações geradas pela máquina da EMPRESA A, é assinalada com as variáveis e as tuplas citadas nos itens 1, 2, 3 e 4 (vide a representação em vermelho na Tabela 9.2). Lembrando sempre que a máquina a possui uma estrutura aberta e deve ser configurada segundo as necessidades da empresa produtora de software.

Máquina de processo				
Cliente – Empresa XYZ (i)(G)				
Projeto – ABC (p)(G)				
Ordens de serviços: Equalização desenvolvida em 00/00/00 (a)(E)	Ordem: 1	Produto – 1.1 Est: Pronto (v)(N)	Envolvidos: José (Líder) João Marcos	Atividade: (a)(E) Cod: Início: 00/00/00 Tér: 00/00/00 h: 3/4 Teste: Início: 00/00/00 Tér: 00/00/00 h: 1/2 (t)(E)
		Produto – 1.2 (v)(N)	Envolvidos: Maria Pedro	Erros encontrados a partir da atividade de teste: (t)(E) Erro A Erro B
	Ordem: 2			
	Ordem: 3			
Legenda: Est: Estado do Produto = Em desenvolvimento ou Pronto h: n/n1 = h – carga horária: n prevista, n1 realizada.				

Tabela 9.2 – Formalização das características das informações geradas pela máquina de processos.

9.4 A Aderência dos Casos às Bases de Dados do Modelo

Ao analisar o modelo preliminar apresentado no capítulo 4, é possível encontrar 4 bases de dados: estrutura da PML; base de projetos; base de processos e base moldes para definição dos ativos do processo.

A base de projeto tem como objetivo armazenar dados sobre os projetos executados, a partir dos processos gerados pelo modelo. Salienta-se que, até o momento, nenhum projeto foi executado⁵. Fato este que inviabiliza, momentaneamente, a verificação de tal base.

⁵ A execução de um processo será apresentada no capítulo 10.

A base estrutural da linguagem de modelagem de processo tem como objetivo armazenar estrutura sintática e semântica da PML. Tal estrutura foi utilizada na modelagem do processo e da atividade de equalização. Ressalta-se que até o momento, somente uma PML está armazenada nesta base, porém outras podem ser inseridas como, por exemplo, a UML (*Unified Modeling Language*).

A base de molde tem como objetivo armazenar a configuração sintática para a definição dos ativos do processo. Um molde genérico para a configuração deles, também, foi armazenado e um ativo foi modelado, satisfatoriamente, com tal configuração. Ressalta-se, também, que tal base está aberta a outros moldes.

Por fim, a base de processo tem como objetivo armazenar os processos executados, juntamente, com máquina de processo. Ressalta-se, novamente, que nenhum processo foi executado. Fato este que inibe, momentaneamente, a verificação de tal item.

9.5 Conclusões Inferidas a partir das Considerações Efetuadas neste Capítulo

Este capítulo procurou verificar a aderência dos casos apresentados nos capítulos 2 e 7 e nos anexos A e B com o modelo preliminar estruturado no capítulo 5.

Ao analisar as informações apresentadas é possível verificar as seguintes alterações no modelo:

- A definição do domínio do conhecimento ou a segmentação do produto está relacionada às questões estratégicas da empresa de produção de software com características fabris.

- A base de moldes alimenta, diretamente, a atividade de instanciação.
- Com base na PML utilizada é visto que: os recursos humanos e ferramental são distribuídos na atividade de modelagem e não na atividade instanciação.

De posse de tais alterações é possível conceber uma nova estrutura visual (vide Figura 9.4) para o modelo preliminar proposto no capítulo 5.

Figura 9.4 - Estrutura de modelo – evolução 1

Por fim, cabe ressaltar que as atividades de simulação, execução, avaliação e armazenamento serão verificadas no capítulo 10 (as atividades de seleção e adaptação não serão verificadas neste trabalho).

Capítulo 10 – Aplicação do Modelo Utilizado para Criação e Organização de Processo de Software: Um Estudo de Caso

Este capítulo tem como objetivo relatar, a partir de um estudo de caso, o comportamento do modelo, proposto preliminarmente no capítulo 5 e alterado parcialmente no capítulo 9, em um ambiente real. Ressalta-se, novamente, modelo pode evoluir ou sofrer algum tipo de alteração, dado o seu comportamento.

A divisão estrutural das seções, utilizada neste capítulo, está baseada no roteiro para o desenvolvimento do estudo de caso nesta etapa, item D da Tabela 6.4.

10.1 Histórico e Descrição do Ambiente para Realização do Estudo de Caso

Em abril de 2005, surgiu à idéia de se implantar o processo de produção de software desenvolvido pelo ELABSOFT na Faculdade de Tecnologia de Ourinhos (FATEC-OU), como parte do projeto de tese discutido neste trabalho. A idéia embrionária era desenvolver um ambiente experimental, na qual os alunos dos cursos de Análise de Sistemas e Tecnologias da Informação pudessem colocar em prática os conceitos abordados nas disciplinas da área engenharia de software¹.

Paralelamente ao fato acima explicitado, o poder público municipal da cidade de Ourinhos desenvolveu um plano migratório de empresas de tecnologia. O objetivo de tal plano é prover incentivos fiscais e recursos humanos às empresas que se dispusessem a se instalar na cidade.

¹ As disciplinas da área de engenharia de software dos cursos de Análise de Sistemas e Tecnologias da Informação da FATEC-OU e de Ciência da Computação da Fundação Educacional do Município de Assis, são permeadas por conceitos de produtividade, qualidade e fábrica de software (BEGOSSO e FILGUEIRAS (2003), FABRI et. al. (2005b e 2006a)).

Durante a etapa de divulgação do plano, às empresas localizadas na grande São Paulo, o diretor da FATEC-OU (envolvido diretamente na concepção do plano), além de apresentar as vantagens e benefícios, salientou que possuía dois professores que trabalhavam, ativamente, com o tema fábrica de software. Sendo que um deles estava desenvolvendo um modelo que possibilitava a criação e a organização de um processo de produção de software com características fabris. Ressaltando que tais professores estavam interessados em coordenar o processo de implantação de tais fábricas dentro da FATEC-OU (fato este ocorrido nos meses de junho e julho 2005). Além disto, o diretor se propunha disponibilizar a infraestrutura necessária para que uma das empresas realizasse um projeto piloto dentro da Faculdade de Tecnologia de Ourinhos, comprometendo assim, os professores que trabalhavam com tal tema. A contra-partida da empresa interessada seria a contratação, de forma remunerada, de alunos da FATEC para trabalhar no projeto.

Tendo em vista a aquisição de benefícios fiscais, proposta pelo plano migratório, a possibilidade de se beneficiar da infra-estrutura proposta pelo diretor da FATEC-OU e a utilização de uma vasta gama de mão de obra da região, uma das empresas se interessou pela idéia.

Os fatos acima explicitados, levaram o autor deste trabalho, com o apoio direção da Faculdade de Tecnologia de Ourinhos, a desenvolver, em outubro de 2005, o projeto de pesquisa “Desenvolvimento e Implementação de um Processo Fabril para Produção de Software em um Contexto Acadêmico”. Tal projeto foi encaminhado a Comissão do Regime de Trabalho Permanente (COPERT) do Centro Estadual de Educação Tecnológica Paula Souza (CEETPS), analisado e aprovado em dezembro de 2005. O professor responsável por tal projeto teve sua jornada de trabalho aumentada de 16 para 40 horas semanais (16 horas/aulas e 24 para o desenvolvimento do projeto).

Em 16 de janeiro de 2006, as atividades propostas no projeto, começaram a ser desenvolvidas. Inicialmente, o autor deste trabalho configurou o ambiente experimental executando as seguintes tarefas:

- Seleção do pessoal. Foram selecionados 4 alunos para trabalhar no projeto de desenvolvimento do software (carga horária: 20 horas/semana para cada aluno²);
- Configuração da infra-estrutura de hardware necessária para execução do projeto. A FATEC-OU instalou um laboratório com 2 computadores, conectados a Internet para o desenvolvimento do trabalho. A empresa migratória disponibilizou toda infra-estrutura para comunicação remota dos envolvidos com o projeto (alunos da FATEC-OU encontravam-se em Ourinhos e gerentes de projetos da empresa encontravam-se em São Paulo);
- Configuração da infra-estrutura de software necessária para execução do projeto. Tal configuração será apresentada na próxima seção.

Em 20 de fevereiro as atividades de configuração do ambiente estavam consolidadas.

10.2 O Levantamento de Requisitos para Definição do Processo de Produção de Software

Com base na representação gráfica do modelo proposta na Figura 9.4 é verificado a necessidade de definição dos requisitos do processo de produção de software. Dentro desta ótica foi estabelecido que:

- O ciclo de produção de fábrica de software da FATEC-OU seria caracterizado como curto (as atividades de equalização, codificação, teste, gestão de projeto e da qualidade do processo e do produto foram estabelecidas). Salienta-se que a definição de tal ciclo se deu em comum acordo entre a empresa migratória e a FATEC-OU;

² A remuneração de cada aluno girava em torno de R\$ 350,00 por mês.

- Na definição do ambiente tecnológico estabeleceu-se a utilização das ferramentas livres. Entre estas é possível destacar a linguagem PHP (WELLING e THOMPSON (2005)) e o banco de dados MySQL (WELLING e THOMPSON (2005)). O aparato ferramental para o planejamento e controle do projeto, também, foi mapeado. A máquina de processo, ferramenta presente em todos os casos delineados no capítulo 7 foi desenvolvida, internamente, na FATEC-OU. Ressalta-se que a máquina, em questão, é utilizada de forma conjunta com o *Gantt Project* (ferramenta de gestão de projeto, caracterizada como livre – maiores informações sobre tal ferramenta consulte <http://ganttproject.sourceforge.net/>). O *Subversion* é a ferramenta de gestão de versão e de configuração utilizada pelo processo. Por fim, na visualização dos artefatos advindos da atividade de projeto de software foi utilizada a ferramenta *Enterprise Architect*³ (EA). Salienta-se que a definição do aparato ferramental para o desenvolvimento de software e controle do projeto se deu em comum acordo entre a empresa migratória e a FATEC-OU;
- Na segmentação de produto, a fábrica de software FATEC-OU iria codificar e testar sistemas de informações automatizados. Dentro desta linha, a empresa migratória, juntamente com a FATEC-OU, optaram por desenvolver um software para a gestão de carreiras dentro de um ambiente de recursos humanos;
- O modelo da qualidade estabelecido na fábrica de software da FATEC-OU estaria alinhado com o CMMI. Ressalta-se que a unidade estabelecida na faculdade não possui recursos financeiros para atingir uma certificação oficial, porém verificações não oficiais podem ser feitas pelos professores e convidados a participar do projeto da fábrica.
- Na definição dos recursos humanos foram selecionados 4 (quatro) alunos para trabalhar no projeto. O processo de contratação ficou sob a responsabilidade da FATEC-OU e foi realizado da seguinte forma:

³ Esta ferramenta é proprietária e foi disponibilizada para FATEC-OU pela empresa migratória.

- a. Publicação do edital de seleção: O edital foi divulgado a todos os alunos;
- b. Recebimento de currículos por parte da secretaria acadêmica: O aluno, interessado em trabalhar no projeto, realizou a sua inscrição no processo de seleção, entregando o currículo na secretaria acadêmica;
- c. Análise do currículo e do histórico dos alunos inscritos: Esta análise foi realizada pelos professores envolvidos com o projeto (ambos com formação na área de engenharia de software). Como critério, foi levando em conta o conhecimento na linguagem PHP e no banco de dados MySQL;
- d. Entrevista com os alunos: Realizadas pelos professores envolvidos com o projeto. Somente os alunos que tiveram o currículo aprovado no item anterior foram convocados para a entrevista;
- e. Aval dos professores envolvidos com o projeto para contratação dos alunos. O aval dos professores foi efetuado com base na análise de currículos, histórico curricular, consulta aos demais professores e desempenho na entrevista;
- f. Contratação dos alunos por parte da empresa migratória⁴.

Além dos conhecimentos ligados à linguagem de programação e o banco de dados utilizados no projeto, os alunos contratados tinham uma base conceitual, advinda da graduação, de UML e gestão de projetos. Conhecimentos sobre métricas de software e sobre a teoria básica de processo tiveram que ser aprofundados. Este fato levou os professores, envolvidos com o projeto, a configurarem dois treinamentos:

⁴ A contratação dos alunos selecionados por parte da empresa migratória foi efetuada em julho de 2006. Já a execução dos itens das demais tarefas foi realizada de 16/01/06 a 20/02/06.

- Treinamento básico em métricas de software, mais precisamente em pontos por função. Treinamento feito com casos reais de projetos de software (duração 16 horas);
- Treinamento na teoria básica de processo de software: Aspectos sobre processo de produção, artefatos de projeto e modelos de processo foram apresentados aos alunos (duração 16 horas);

Na definição de tais treinamentos os professores envolvidos com a implantação da fábrica de software utilizaram o modelo de preservação de conhecimento proposto por TRINDADE (2006) (vide Figura 10.1).

Figura 10.1 – Modelo de preservação da conhecimento. Adaptado de TRINDADE (2006)

Segundo TRINDADE (2006), o modelo tem como base a participação do aluno (os aprendizes) e do instrutor (os professores) em um processo de ensinagem (termo utilizado pelo autor). Este processo produz um conhecimento de grau N, em nosso caso conhecimentos relacionados aos conceitos de métricas e processo de produção de software. O modelo, também, prevê a geração da documentação sobre o processo de ensinagem. Ressalta-se que os alunos treinados, sob a ótica do modelo, podem utilizar os documentos armazenados no repositório e se caracterizarem como instrutores, quando novos alunos forem contratados.

Definidos os requisitos estabelecidos no modelo, as próximas seções apresentarão a execução das atividades do modelo.

10.3 A Atividade de Modelagem

Conforme relatado nos capítulos 5 e 9, a modelagem do processo de produção de software deve ser elaborada utilizando uma PML. O modelo, prevê a utilização do atigrama definido pela normalização IDEF-0. Com base nestes argumentos será modelado, a título ilustrativo, somente a visão processual da estrutura de produção da fábrica de software em questão (vide Figura 10.2). Salienta-se que a modelagem das atividades da visão gerencial, em nosso caso gestão de projetos e da qualidade, segue a mesma sistemática da modelagem da visão processual.

As considerações relacionadas às perspectivas de fluxo de trabalho, fluxo de dados, recursos e habilidades efetuadas na seção 9.2, também, são validas para esta seção.

Figura 10.2 – Atividades do processo de produção da fábrica de software da FATEC-OU (visão processual).

Ao analisar a Figura 10.2 é possível notar a aderência do processo de produção da FATEC-OU ao ciclo curto de produção de software apresentado no capítulo 2. Os artefatos de entrada e de saída, as regras para desenvolvimento das atividades (que também podem ser configuradas com algum artefato), os envolvidos e o ferramental necessário, para a execução do processo, também, estão explicitados na figura.

10.4 A Atividade de Instanciação

Conforme relatado na seção 9.3, a atividade de instanciação parte das informações geradas na atividade de modelagem e concebe as tarefas e os ativos do processo. É importante ressaltar que, a título ilustrativo, somente as tarefas da atividade de codificação serão instanciadas, as demais seguem a mesma sistemática.

Na atividade de instanciação, também, é prevista a alocação dos recursos humanos e tecnológicos junto às atividades do processo, porém tal alocação foi realizada na atividade de modelagem, este fato contraria o modelo definido no capítulo 5 e, com certeza, provocará uma alteração no mesmo.

Por fim, a máquina de processo, utilizada pela fábrica de software da FATEC-OU será descrita, integralmente.

10.4.1 Definindo Tarefas e Ativos do Processo

A atividade de codificação do processo de produção da fábrica em questão possui as seguintes tarefas:

- Tarefa um (T1): Receber ordens de serviço.
 - Objetivo: Programador acessa a máquina de processo e visualiza a ordem de serviço (OS) a ser codificada.

- Tarefa dois (T2): Codificar a OS.
 - Objetivo: Codificar uma determinada OS respeitado o código de convenção JAVA (JAVA Code Convention (tal código pode ser consultado na íntegra em <http://java.sun.com/docs/codeconv/>)) e a linguagem de programação estabelecida no projeto.
- Tarefa três (T3): Alimentar Máquina de Processo.
 - Objetivo: Após a codificação, o programador alimenta a máquina de processo com informações relacionadas ao tempo de produção e recursos consumidos.

De posse da descrição das tarefas, é possível modelá-las utilizando a PML descrita no capítulo 5 (vide Figura 10.3).

Ao analisar tal figura é possível perceber a presença dos seguintes ativos do processo (ou artefatos): ordem de serviço e o código de convenção JAVA. Para fins ilustrativos, é apresentada no Quadro 10.1, a estrutura de uma ordem de serviço emitida pela máquina de processo.

Figura 10.3 - Modelagem das tarefas da atividade de codificação. Processo fabril da FATEC-OU

Ao analisar o Quadro 10.1 é possível perceber que a desenvolvedora Adriana Figueiredo Pereira, está envolvida como projeto ABC, tendo como responsabilidade desenvolver o componente (ou ativo de software) denominado como cadastro de idiomas. Tal componente se encontra na versão 5.1 e sua especificação, artefatos desenvolvidos pela atividade de projeto de software encontram-se disponíveis no endereço <http://192.168.101.3/modelagem/>. A versão 5.1 foi concebida em 07/07/2006 e está esperando para ser codificada. Ao codificar tal versão, Adriana deve informar a data, a hora de início e hora de término da atividade.

Quadro 10.1 - Apresentação do artefato ordem de serviço – gerada pela máquina de processo – Fábrica de software da FATEC-OU

Faculdade de Tecnologia de Ourinhos - Laboratório Fábrica de Software (LFS)			08/11/2006 16:00
Ordem de Serviço para Execução			
Funcionário:	3	Adriana Figueiredo Pereira	
Ativo Software:	5	Cadastro de Idiomas	
Tipo:	1	Componente de manutenção de dados	
Nome do Projeto:	ABC		
Código da Versão:	5.1		
Especific. Versão:	http://192.168.101.3/modelagem/		
Data Início Versão:	7 /7 /2006		
Data TérminoVersão			
Atividade:	1	Codificação	
Data Início Ativ.:	7 /7 /2006		Hora Início Ativ.:
Data Término Ativ.:			Hora Término Ativ.:

10.4.2 Estruturando a Máquina de Processo

A estruturação da máquina de processo, utilizada pela FATEC-OU, foi efetuada segundo a formalização apresentada no capítulo 5. A concepção de tal máquina levou em consideração:

- O armazenamento das informações do projeto de concepção do ativo de software (ou componente) – a gestação;
- O armazenamento das informações inerentes à versão 1 do componente – o nascimento;
- O armazenamento das informações geradas com a execução das atividades do processo – o crescimento;
- O armazenamento das informações sobre a concepção de um componente a partir da união de dois ou mais componentes – a reprodução;
- O armazenamento das informações que caracterizam o óbito do componente, por exemplo, a sua morte tecnológica.

O armazenamento das informações citadas pode ser constatado na Figura 10.4. Ao analisar esta figura é possível verificar, destacado em *itálico>, a correlação entre o modelo de dados da máquina e as tuplas G , N , E , R e a variável M , definidas no capítulo 5.*

Figura 10.4 - Organização da máquina de processo

Efetuada uma análise, mais aprofundada da Figura 10.4, é possível verificar que o "gerente de projeto" tem acesso à toda "história de vida" de qualquer componente ou ativo. Uma das possíveis informações que a máquina fornece pode ser verificada por meio da Figura 10.5.

Figura 10.5 - História de vida do componente

Na Figura 10.5 é perceptível que:

- o “componente A” possui "Registro Geral (RGC) = 001-c/95”;
- foi "concebido" pelo “projeto Y” da “empresa XYZ”;
- o componente "nasceu" em “16/01/2005”;
- iniciou seu "crescimento" no dia “17”;
- e o mesmo continua em "evolução" (Q = E – vide tuplas e variáveis no capítulo 4);
- o componente “001-c/95” é "filho" dos componentes “008-c/94” e “009-c/94”.

Concebidas as tarefas, modelados os ativos de processo, a proposta preliminar estabelece que os recursos, ferramental e humano, para a implantação do processo sejam alocados. Neste momento o cronograma para implantação do processo é desenvolvido (vide Figura 10.6).

Duração das tarefas a serem desenvolvidas no processo de implantação

Alocação de recursos humanos na execução das tarefas

Figura 10.6 – Cronograma e definição dos recursos para implantação do processo de produção de software.

Ao analisar a Figura 10.6 é perceptível que a implantação do processo, envolvendo a atividade de simulação, teve uma duração de 7 semanas.

10.5 A Atividade de SIMULAÇÃO

Conforme descrito no capítulo 5, a atividade de simulação do modelo proposto tem como objetivo estimar a duração do processo, quando este estiver em execução, e verificar a ocorrência de possíveis problemas.

A título ilustrativo, esta seção apresentará a simulação das atividades de equalização, codificação e teste de um componente do “software de gerenciamento de dados de empréstimos e consultas da biblioteca da FATEC-OU⁵”.

Na atividade de simulação, os requisitos sistêmicos foram capturados pelo autor deste trabalho, e modelados com Linguagem Unificada de Projeto (*Unified Modeling Language – UML*)⁶. Um dos artefatos gerados, com a linguagem em questão, é o digrama de seqüência⁷ (LARMAN (2000)). Um fragmento do diagrama gerado pode ser verificado por meio da Figura 10.7.

Figura 10.7 – Diagrama de seqüência gerado na atividade de projeto de software.

A Figura 10.7 materializa a especificação da funcionalidade de software “consultar aluno” e, textualmente, pode ser especificado da seguinte forma: A interface do software provê mecanismos para que a Bibliotecária informe o registro acadêmico do Aluno. De posse do registro acadêmico o software consulta a base de

⁵ O software foi utilizado como objeto de simulação do processo e em linhas gerais, o seu objetivo do software é proporcionar aos alunos, professores e funcionários a consulta, via Internet, do acervo da biblioteca da Faculdade de Tecnologia de Ourinhos. Além de possibilitar a reserva de um exemplar para, posteriormente, efetuar o empréstimo.

⁶ Tendo em vista que o objetivo deste trabalho é propor uma estrutura (ou modelo) que possibilite a criação ou organização de um processo de produção de software, o documento de especificação do projeto de software, utilizado na atividade de simulação não será, totalmente, contemplado no texto.

⁷ O diagrama de seqüência tem como objetivo apresentar a relação temporal dos objetos que compõem um software.

dados, inserida dentro do Sistema de Controle Acadêmico, disponibilizando os dados cadastrais do aluno para a visualização da bibliotecária.

Ressalta-se, ainda, que as linhas tracejadas, apresentadas na Figura 10.7, representam a questão temporal do diagrama e os símbolos localizados na parte superior apresentam a relação dos objetos.

De posse dos artefatos do projeto de software, a atividade de equalização entra em execução. Lembrando que nesta atividade os desenvolvedores, verificam questões relacionadas a “corretude” e “completude” das ordens de serviço. Alguns documentos são gerados com a equalização, entre eles é possível destacar a ata de equalização. Nela são registradas as possíveis dúvidas em relação à compreensão dos artefatos.

A ata em questão é preenchida pelos desenvolvedores (neste caso, os alunos envolvidos com o projeto) no momento de equalização e enviada, via e-mail, aos projetistas (neste caso, os dois professores envolvidos com o projeto). Os projetistas respondem, formalmente via e-mail, as dúvidas contempladas na ata. Os projetistas, também, podem solicitar uma reunião com os desenvolvedores, se julgarem necessário.

Conforme estabelecido na Figura 10.2, realizada a equalização, as próximas atividades a serem executadas são: codificação e testes. Tais atividades são delegadas aos desenvolvedores pelo gerente de projeto (em nosso, um dos professores representou este papel). Este fato pode ser verificado na Figura 10.8 e na Figura 10.9.

Segundo a OS, espelhada na Figura 10.8, é perceptível que a funcionalidade cadastrar aluno deverá ser codificada em 15/05/2006 pelo desenvolvedor Henrique Minoru Yamaji, a partir das 8:00. Henrique deverá concluir esta tarefa em 16/05/2006 para que o testador, Daniel Sales dos Santos, inicie a atividade de teste em 17/05/2006. A previsão de término para esta atividade é 18/05/2006.

Ao concluir tais atividades os envolvidos com o processo informam a data e hora de término da atividade e a quantidade de horas gastas para desenvolvimento das mesmas. Salienta-se que estas informações são inseridas na máquina de processo.

Faculdade de Tecnologia de Ourinhos - Laboratório Fábrica de Software (LFS)			
		Ordem de Serviço para Execução	
14/5/2006 16:30		Pág. 1 de 1	
Funcionário:	5 Henrique Minoru Yamaji		
Ativo Software:	13 Consultar Informações de Aluno		
Tipo:	5 Consulta		
Nome do Projeto:	Software para Gestão de Biblioteca		
Código da Versão:	SGBCA1		
Específic. Versão:	\\projeto\SGB\FATEC-OU\ESP		
Data Início Versão:	15/5/2006		
Data Término Versão:			
Atividade:	1 Codificação		
Data Início Ativ.:	15/ 5/2006	Hora Início Ativ.:	08:00:00
Prev. Término Ativ.:	16/5/2006	Prev. Término Ativ.:	
Data Término Ativ.:		Hora Término Ativ.:	

Figura 10.8 – Ordem de Serviço para codificação (gerada pela máquina de processo na atividade de simulação)

Faculdade de Tecnologia de Ourinhos - Laboratório Fábrica de Software (LFS)			
		Ordem de Serviço para Execução	
14/5/2006 16:31		Pág. 1 de 1	
Funcionário:	6 Daniel Sales dos Santos		
Ativo Software:	13 Consultar Informações de Aluno		
Tipo:	5 Consulta		
Nome do Projeto:	Software para Gestão de Biblioteca		
Código da Versão:	SGBCA1		
Específic. Versão:	\\projeto\SGB\FATEC-OU\ESP		
Data Início Versão:	15/5/2006		
Data Término Versão:			
Atividade:	3 Teste		
Data Início Ativ.:	17/ 5/2006	Hora Início Ativ.:	08:00:00
Prev. Término Ativ.:	18/5/2006	Prev. Término Ativ.:	
Data Término Ativ.:		Hora Término Ativ.:	

Figura 10.9 – Ordem de serviço para teste (gerada pela máquina de processo na atividade de simulação)

Já a atividade de teste contempla as seguintes tarefas: projetar casos de testes; gerar dados de testes; testar e; preencher relatório.

No desenvolvimento da atividade em questão o testador utiliza um artefato denominado “*checklist* de teste”. Tal artefato está representado na Tabela 10.1.

Teste de Caixa branca					
O que testar			Correto		Incorreto
Todas as variáveis de programa são iniciadas antes de seus valores serem utilizados?			X		
Todas as constantes foram denominadas?			X		
O limite superior de vetores devem ser igual ao tamanho do vetor ou ao tamanho -1?			X		
Se as <i>strings</i> de caracteres são utilizadas, um delimitador é explicitamente designado?			X		
Existe alguma possibilidade de <i>overflow</i> de buffer?			X		
Teste de Caixa Preta (Dados)					
Casos de testes	Dados de Testes		Resultados	Relato do Caso	
1 - Teste de consistência, exemplo um campo deve possuir somente entrada SIM ou NÃO, entre com outra informação.	Campo	Valores	Saída	Resultado esperado	st
				[C] – correto [I] – incorreto	
2 - Campos que estão gerando o dígito verificador.	Campo	Valores	saída	Resultado esperado	st
	RA	051096-2	Não deve aceitar entrada pois o dígito do RA em questão é 3	Mensagem de Erro	I
3- Entrada com valores superior ao limite do campo.	Campo	Valores	saída	Resultado esperado	st
	RA	9999999999	Não deixa efetuar entrada	Mensagem de Erro	C
4 - Entrada com número negativo.	Campo	Valores	Saída	Resultado esperado	st
	RA	-1	Não deixa efetuar entrada	Mensagem de Erro	C
5 - Entrada com valor <i>ZERO</i> ou <i>BRANCO</i> .	Campo	Valores	saída	Resultado esperado	st
	RA	0	Não deixa efetuar entrada	Mensagem de Erro	C
6 - Troca de tipo de dados.	Campo	Valores	Saída	Resultado esperado	st
	RA	AAAAA-A	Não deixa efetuar entrada	Mensagem de Erro	C
Teste de Interface					
O que testar	Correto?		Modificações		
	Sim	Não			
Campos estão bem definidos na tela?	X				
Objetos estão habilitados corretamente?	X				
Aperte os botões habilitados em situações que não seriam próprias?	X				
Botões que funcionam como links estão abrindo as páginas ou formulários que deveriam abrir?	X				
O acesso a outras ferramentas está acontecendo corretamente, por exemplo: (o botão que acessa a calculadora está realmente chamando a mesma)?	X				

Tabela 10.1 - Checklist de teste – (Caixa Branca, Caixa Preta e de Interface). Dados gerados com a simulação do processo.

Ao analisar a tabela em questão é possível verificar que a fábrica de software da FATEC-OU segue o mecanismo de teste de caixa branca, caixa preta e de interface. Um maior detalhamento sobre estes tipos de testes pode ser encontrado em SOMMERVILLE (2003).

Nota-se que após a execução da atividade de teste, as funcionalidades implementadas são disponibilizadas para os projetistas, na qual estes emitem o parecer favorável ou desfavorável para com elas.

Ressalta-se que a atividade de simulação gerou vários ajustes no processo e na máquina, entre eles é possível destacar:

- A inserção da especificação da interface como regra na atividade de codificação;
- A inserção das informações relacionadas à data e hora prevista para término das atividades espelhadas nas ordens de serviços;
- Configuração de vários relatórios gerenciais para o controle do processo e do projeto. Uma sistematização arquitetônica destes relatórios esta expressa na Figura 10.5.

10.6 A Atividade de EXECUÇÃO

Na atividade de execução (iniciada em julho de 2006), o processo simulado entra em operação sobre um projeto real, a fim de gerar um produto caracterizado como software.

O software a ser desenvolvido pela fábrica da FATEC-OU deve possuir as características apresentadas no Quadro 10.2.

Quadro 10.2 – Características do software a ser desenvolvido pela FATEC-OU (Atividade de execução)

• O cadastro de Colaboradores da empresa migratória, é realizado através de uma planilha de calculo, dificultando uma consulta ágil e fácil. Algumas limitações são encontradas neste cadastro: • Falta de agilidade na captação de currículos; • Dificuldade na consulta de profissionais; • Dificuldade em identificar competências na alocação / preenchimento de vagas; • Falta de histórico dos profissionais; • Dificuldade de reaproveitamento de candidatos para diferentes vagas com o mesmo perfil. • Solução proposta: • Desenvolver uma aplicação na plataforma WEB, capaz de atender os seguintes objetivos: ○ Agilizar a captação de profissionais e currículos; ○ Centralizar informações de colaboradores, ex-colaboradores, <i>free lancers</i> e candidatos; ○ Disponibilizar uma base de dados com uma consulta ágil e fácil, com os dados de profissionais e competências.

É importante ressaltar que as características apresentadas foram retiradas, na íntegra, do documento de Especificação de Requisito de Software (ERS) gerado pela empresa migratória (a partir deste momento denominada como EMPRESA X).

Além das características, a ERS apresenta os requisitos funcionais da aplicação a ser codificada e testada pela fábrica de software da FATEC-OU. Tal especificação pode ser verificada no anexo C deste trabalho:

Por fim, a EMPRESA X, também, disponibilizou para a fábrica de software da FATEC-OU os seguintes artefatos de projetos:

- Casos de uso;
- Diagramas de sequência;
- Diagrama de Classes;
- Modelo de entidade e relacionamento (modelo de dados);
- Modelo de Interface;
- Cronograma global para desenvolvimento do produto.

É importante ressaltar que o autor do trabalho não obteve uma autorização formal da EMPRESA X para divulgar os artefatos acima citados.

10.6.1 Execução da Atividade de Equalização

De posse dos artefatos de projeto, é possível iniciar a fabricação do produto, via atividade de equalização. A comunicação na atividade em questão é realizada, na maior parte das vezes, via e-mail, um exemplo deste fato pode ser comprovado

ao analisar o Quadro 10.3, o qual apresenta uma dúvida (real) dos desenvolvedores, em relação ao modelo de dados e a resposta emitida pelo gerente de projeto.

Quadro 10.3 – E-mail estabelecendo a atividade de equalização.

<p>[E-MAIL COM DÚVIDA] Boa Tarde Cassius.</p> <p>Estávamos estudando o MER e verificamos a necessidade de algumas alterações. Por isso estamos enviando um novo com as modificações com base nas informações que recebemos na ERS. Existe apenas uma dúvida e esta está espelhada nas entidades de cor diferente. Entendemos que tal entidade gera uma única tabela, denominada como descrição da função. Seria isso mesmo?</p> <p>Abraços. Adriana, Daniel, Débora, Henrique</p> <p>[E-MAIL DE RESPOSTA] Pessoal,</p> <p>A idéia é mais ou menos essa mesmo, do conjunto dos registros vindos destas entidades formarem a descrição da função. Temos algumas considerações quanto a algumas entidades, que vamos ter que resolver aqui e informá-los logo em seguida. Notem no anexo que boa parte do modelo está em outra cor, que são as entidades que consideramos como já "resolvidas". As que estão em branco ainda temos algumas dúvidas e vamos analisar com mais calma. Deixamos uma em cinza, que demonstra a idéia que temos da diferenciação de um treinamento realizado e um treinamento requerido para uma função. Quanto a iniciar o desenvolvimento, vamos passar uma primeira versão do cronograma amanhã. Mas já será possível visualizar sua próxima tarefa, que é o protótipo navegável (início da navegação). Depois virá a implementação do código do software.</p> <p>E os <i>use-cases</i>, tiveram dúvidas? Podemos trabalhar com este formato? A partir deles vocês conseguem gerar a especificação técnica, para codificação?</p> <p>Um abraço, Cassius</p>

Um dado interessante, pertinente à atividade de equalização, é o número de e-mails trocados entre os desenvolvedores e o gerente de projeto. De 04 de julho a 25 de outubro foram trocados 42 (salienta-se que a maioria dos e-mails possuem mais que uma dúvida). Já as reuniões via VoIP (voz sobre IP) chegaram à casa de 10.

Por fim, é importante salientar que a atividade de equalização sofreu algumas alterações, durante a execução do processo, em relação à sua simulação. Tais alterações estão destacadas, em negrito, na Tabela 10.2.

Atores	Responsabilidades		
Gerente de projetos (EMPRESA X) Realiza o papel do projetista na simulação do processo.	Disponibilizar os seguintes artefatos de projeto de software para codificação. Especificação de requisitos de software; Modelo de dados; Casos de Uso; Diagramas de Sequência; Diagrama de Classes. Cronograma global de projetos; Modelo de interface; (Tais artefatos não eram disponibilizados, formalmente, aos desenvolvedores no ambiente simulado). Negociar junto ao Mediador de Projetos da FATEC-OU questões relacionadas ao cronograma. (Esta negociação não estava presente no ambiente simulado, visto que o projeto não necessitava de um gerenciamento global). Sanar as dúvidas que os desenvolvedores tenham sobre os artefatos enviados (equalizar). As dúvidas devem ser sanadas em um primeiro momento via e-mail. Se necessários às dúvidas devem ser sanadas via VOIP (dispositivo de voz sobre IP). (Este último meio de comunicação não foi contemplado na simulação do processo, visto que as reuniões, entre desenvolvedores e projetistas, foram realizadas presencialmente).		
Mediador de projetos (FATEC-OU): Este papel não estava contemplado no processo simulado, na FATEC-OU papel é desempenhado pelos professores, caracterizados como coordenadores do projeto.	Negociar questões relacionadas a cronograma junto ao Gerente de Projetos da <u>EMPRESA X</u> . Atribuir as ordens de serviços para os Desenvolvedores. Acompanhar a produtividade em horas dos Desenvolvedores. Orientar os Desenvolvedores no desempenho de suas atividades. Disponibilizar para o Gerente de Projetos a produtividade de cada desenvolvedor ou da célula de desenvolvimento. (Este item, também, não foi contemplado no processo simulado, visto que as tarefas, inerentes ao gerenciamento do projeto, eram de responsabilidade dos projetistas (no caso da simulação, os professores)).		
Desenvolvedores:	Compreender os artefatos enviados pelo Gerente de Projetos. Comunicar-se com o Gerente de Projetos sempre que for necessário. Em um primeiro momento via e-mail. Quando solicitado, a comunicação deve ser estabelecida via VOIP (solicitação esta efetuada pelo Gerente de Projetos) Codificação das ordens de serviços estabelecidas pelo Mediador de Projetos. Desenvolver o teste unitário. Disponibilizar unidade de software para o teste global. Apontar as horas em relação à ordem de serviço.		
DESCRIÇÃO DO PROCESSO			
Entradas	Ações a serem executadas	Saídas	Envolvidos
Artefatos de projetos	1 - Gerente de Projeto disponibiliza artefatos (via sub-version). 2 - Desenvolvedor informa recebimento dos artefatos (via e-mail). 3 - Desenvolvedor solicita esclarecimentos sobre os artefatos. 4 - Gerente de Projetos comunica o recebimento da dúvida e tempo que levará para saná-la. 5 - Gerente de Projetos sana as dúvidas. 6 - Gerente de Projetos estabelece uma comunicação via VOIP, se necessário.	Dúvidas sobre os artefatos. Aviso sobre entendimento dos artefatos. Ata de Equalização	Gerente de Projeto Desenvolvedor Mediador

Tabela 10.2 – Configuração da atividade de equalização durante a EXECUÇÃO do processo.

10.6.2 Execução da Atividade de Codificação

Conforme relatado, anteriormente, a atividade de codificação possui as seguintes tarefas: receber ordens de serviço; codificar a OS e alimentar máquina de processo.

A partir dos artefatos equalizados, a ordens de serviços são inseridas na máquina de processo (um exemplo da funcionalidade da máquina de processo foi apresentada na Figura 10.8) e codificadas pelos desenvolvedores. Ao analisar o Quadro 10.3 é possível perceber que a atividade de codificação foi dividida em duas fases: 1) definição de um protótipo navegável e; 2) implementação – as palavras grafadas em negrito no quadro sustentam tal afirmação. Salienta-se que esta divisão não foi contemplada nas atividades de modelagem, instanciação e simulação.

A Figura 10.10 apresenta a execução da tarefa 1 da atividade de codificação. Nela é perceptível que a Débora Cristina de Jesus da Silva recebe uma OS e sua responsabilidade é desenvolver o “cadastro de idiomas”. O início da atividade de codificação é 07/07/2006 e prazo para entrega é 11/07/2006. Débora disponibilizou o produto para teste em 10/07/2007, às 10 horas, levando 4 horas e 30 minutos para desenvolver o cadastro de idiomas (vide data de término da atividade grafada em tinta azul). Os dados manuscritos serão inseridos na máquina de processo, fato delineado na tarefa 3 da atividade em questão. Ressalta-se que a ordem de serviço propõe o desenvolvimento do código e não o do protótipo navegável.

Funcionário:	4 Débora Cristina Jesus da Silva		Pág. 1 de 2
Ativo Software:	5 Cadastro de Idiomas		
Tipo:	1 Componente de manutenção de dados		
Nome do Projeto:	7skill		
<hr/>			
Código da Versão:	5.1		
Especific. Versão:	http://192.168.101.3/modelagem/		
Data Início Versão:	7/7/2006		
Data Término Versão			
<hr/>			
Atividade:	1 Codificação		
Data Início Ativ.:	7/ 7/2006	Hora Início Ativ.:	
Prev. Término Ativ.:	11/ 7/2006	Prev. Término Ativ.:	
Data Término Ativ.:	10/07/2006	Hora Término Ativ.:	10h - 4h 30 min

Figura 10.10 – Acesso real de um desenvolvedor a uma ordem de serviço

10.6.3 Execução da Atividade de Teste

Executada a atividade de codificação, a próxima tarefa a ser desenvolvida é a de teste. Como relatado, anteriormente, as informações relacionadas a tempo e recursos humanos alocados para o desenvolvimento desta tarefa, também, estão inseridas na máquina de processo.

Ao testar um ativo de software o testador informa com base no *checklist* de teste, se o ativo em questão está correto ou incorreto. Caso o ativo esteja incorreto, uma nova OS, classificada como ordem de serviço corretiva, é gerada e o ativo volta à bancada de desenvolvimento, acompanhado de seu *checklist*. As alterações são realizadas e uma nova ordem de serviço de teste é concebida.

Além do teste de caixa preta, caixa branca e de interface, todo ativo de software, gerado pela fábrica da FATEC-OU, é homologado pelos mediadores para, posteriormente, serem entregues. Este fato está, intimamente, ligado ao conceito da qualidade do produto.

Já na atividade de gestão de projetos, tanto o mediador como o gerente de projetos conseguem extrair as seguintes informações da máquina para realizar o seu trabalho:

- Desenvolvedores atrasados em suas atividades;
- Atividades atrasadas e concluídas por projetos;
- Número de ordens de serviço corretivas geradas por projeto e por ativo.

Salienta-se que a máquina de processo possui uma grande portabilidade na questão de extração de dados, com isto vários *layouts* de relatórios podem ser concebidos.

Por fim, a gestão da qualidade do processo é realizada pelos mediadores do projeto (os professores), que verificam, constantemente, se os desenvolvedores estão respeitando as regras estabelecidas.

10.7 A Atividade de AVALIAÇÃO

Esta seção abrigará informações sobre a atividade de avaliação dinâmica do processo. Conforme citado no capítulo 5, este tipo de avaliação ocorre com o processo em execução, e seu principal objetivo é corrigir algum problema ou erro no processo, não percebido na atividade de simulação⁸.

Durante a execução do processo alguns ajustes foram realizados, entre eles é possível destacar:

- Inserção do papel do mediador de projeto (vide Tabela 10.2);
- Agregação de um cronograma global ao projeto, visto que a fábrica de software da FATEC-OU ficou responsável pela codificação e teste do produto software e a

⁸ Salienta-se que no momento da configuração deste trabalho o processo não se encontrava finalizado, fato este que inibe a inserção de informações sobre a avaliação estática.

EMPRESA X foi caracterizada como uma unidade contratante da fábrica de software em questão;

- Reunião com o objetivo de eqüilizar os artefatos e as ordens de serviços passaram a ser realizadas a distância, este fato levou a configuração de uma infra-estrutura de VOIP (voz sobre IP);
- A atividade de codificação foi dividida em duas partes, desenvolvimento da interface (protótipo navegável) e produção do código do software (vide Quadro 10.3). Com isto o modelo de processo foi alterado de incremental para um misto de evolucionário e incremental.

Com base nas alterações delineadas, é possível afirmar que a atividade de avaliação dinâmica estabelece a adaptabilidade do processo gerado pelo modelo ao ambiente de execução. Sem esta atividade seria impossível realizar uma aderência total das atividades simuladas a um projeto real.

10.8 – As Atividades de ARMAZENAMENTO, SELEÇÃO e ADAPTAÇÃO

As atividades de seleção e adaptação não serão apresentadas neste trabalho, visto que o processo, ainda, encontra-se em execução sobre um projeto determinado. Além disto, o processo em questão é o primeiro gerado pelo modelo. Estes fatos inibem a utilização da visão de reuso apresentada na Figura 5.5.

Já a atividade de armazenamento ocorre em paralelo à execução do processo, os dados gerados nas atividades de projeto de software (artefatos disponibilizados para codificação), codificação e teste são armazenados nas bases de dados espelhadas na Figura 9.4.

10.9 – Conclusões inferidas a partir das Considerações Efetuadas neste Capítulo

Este capítulo apresentou um estudo de caso que relata o comportamento prático do modelo proposto, preliminarmente do capítulo 5 e alterado no capítulo 9 (vide Figura 9.4). De posse das informações geradas com o experimento é possível verificar uma nova evolução do modelo, na qual se destacam as seguintes alterações:

1. Os atores, quando selecionados devem ser submetidos a um processo de treinamento. Para realizar este treinamento, foi utilizado o modelo de ensinagem proposto por TRINDADE (2006).
2. O modelo de ensinagem proposto por TRINDADE (2006) gerou uma nova base de dados, denominado como repositório de preservação de conhecimento.
3. O processo de seleção dos atores foi efetuado em um ambiente acadêmico. Para realizar tais tarefas os professores, responsáveis pelo desenvolvimento do projeto levaram em consideração informações do histórico escolar, do currículo, entrevista e consulta aos demais professores. Se o processo de seleção fosse efetuado dentro de um ambiente empresarial, a consulta aos professores não poderia ser definida como um critério de seleção.
4. A atividade de simulação, também, pode ser vista como um treinamento e nada impede que o modelo proposto por TRINDADE (2006) seja aplicado a ela.
5. A atividade de armazenamento não alimenta, somente, a base de processo, mas também a base de projetos.
6. A base de moldes deve ser adaptativa. Algumas das informações presente em tal base, como as representadas no capítulo 5, às vezes não são configuradas nos

artefatos, um exemplo disto é verificado ao se analisar a OS apresentada na Figura 10.8.

Figura 10.11 – Estrutura do modelo – evolução 2

7. A atividade de instanciação será interpretada como um mecanismo de validação e refinamento dos recursos humanos e tecnológicos, a alocação de tais recursos foi realizada pela atividade de modelagem.

De posse de tais alterações é possível conceber uma nova estrutura visual (vide Figura 10.11 números destacados em vermelho) para o modelo.

11 – Conclusões e Trabalhos Futuros

Este capítulo é dedicado às discussões finais sobre a pesquisa e como ela contribuiu para os aspectos produtivos de uma fábrica de software. O presente capítulo está dividido em 3 partes:

- a primeira delas, retrata as contribuições do trabalho, junto aos pilares acadêmico (ou universitário) e empresarial. A verificação do valor verdade¹ das proposições delineadas no capítulo 6, também, é apresentada nesta parte – seções 11.1, 11.2 e 11.3;
- a segunda, direciona os resultados inferidos a partir da relação do modelo proposto e o referencial bibliográfico apresentado nos capítulos 2 e 3 – seção 11.4.
- Por fim, a terceira apresenta as pesquisas futuras que podem ser inferidas a partir deste trabalho – seção 11.5.

11.1 Contribuições do Trabalho junto ao Pilar Acadêmico

Uma das preocupações do trabalho, destacada na introdução, é prover um mecanismo que possibilite o aumento da produtividade do setor de software com qualidade. Para que isto ocorra, é necessário um esforço conjunto do governo, das empresas e da universidade. Ao analisar a organização textual deste trabalho é perceptível que o mesmo ataca duas das três entidades citadas, universidade e empresas.

Dentro do âmbito universitário ou acadêmico o trabalho contribuiu com os seguintes pontos:

¹ O termo valor verdade, advém da lógica aristotélica, na qual uma proposição pode assumir dois valores, verdadeiro ou falso.

1. Organização do modelo de fábrica de software do ELABSOFT. O trabalho apresenta todo o contexto evolutivo do modelo e o embasamento bibliográfico direcionador de tal ponto. Estas informações podem ser encontradas no capítulo 3.
2. Compilação teórica sobre as estruturas que possibilitam a criação e/ou organização de um processo de software. Vários modelos de reuso de processo e a base conceitual sobre meta-processo podem ser verificados no capítulo 4.
3. Proposição preliminar do modelo para a criação, organização, instanciação e adaptação de processos de produção de software para o contexto fabril.
4. Apresentação do processo de produção de software de seis fábricas brasileiras. Este ponto é delineado pelo capítulo 7, de forma condensada, e aprofundada no anexo A. Dentro da perspectiva estrutural do trabalho, as fábricas foram mapeadas sobre os seguintes fatores:
 - a. Caracterização (ou descrição) da empresa pesquisada.
 - b. Definição do processo de produção de software sob a ótica dos ciclos longo, médio e/ou curto.
 - c. Definição dos aspectos processuais, segundo a suas atividades, modelo, envolvidos (atores que atuam diretamente com o processo) e características de utilização (o processo é caracterizado como leve ou pesado).
 - d. Mapeamento das atividades de gestão de configuração, da qualidade e de projetos, incluindo as métricas utilizadas.

- e. Verificação das ferramentas utilizadas para automatizar o processo de produção nas empresas pesquisadas. Neste item, também, foi verificada se as empresas possuem uma biblioteca de componentes organizada e uma base histórica de projetos.
 - f. A orientação das empresas nas questões relacionadas a domínio do conhecimento, a tecnologia e a produto.
 - g. E, por fim, a forma de criação e/ou organização do processo de software. As empresas utilizaram alguma estrutura formal para criar o seu processo de produção ou o mesmo foi criado a partir das experiências dos profissionais que nela atuam.
5. Apresentação do processo de produção de software de cinco empresas estrangeiras (SDC (americana), Hitachi, Toshiba, NEC e Fujitsu (nipônicas)) de forma condensada no capítulo 2 e aprofundada no anexo B. Este ponto foi baseado na obra de CUSUMANO (1991) e é, parcialmente, aderente aos fatores acima delineados.
6. Análise comparativa entre o processo de produção de software das empresas brasileiras e estrangeiras. Esta análise é, totalmente, baseada nas considerações efetuadas no capítulo 2. Além da relação, os casos apresentados contribuíram para a evolução do modelo proposto pelo ELABSOFT.
7. Aderência do modelo proposto no capítulo 5 aos casos apresentados nos capítulos 2 e 7 e nos anexos A e B, também, é um dos pontos a ser destacados neste trabalho. A aderência foi verificada nas “meta-atividades” de modelagem e instanciação.

Antes de iniciar as considerações sobre as contribuições deste trabalho junto ao pilar empresarial, gostaria de estabelecer um diálogo mais aprofundado sobre o ponto 6. Na análise comparativa apresentada, é possível verificar que dentro da

ótica temporal, é perceptível que as considerações efetuadas por CUSUMANO (1991), sobre as fábricas estrangeiras, são datadas da década 1980. Já as empresas brasileiras, excluindo a B e a D, foram fundadas no início da década de 1990. O processo fabril de tais empresas, incluindo as excluídas, foi caracterizado após 1995, anos depois das empresas estrangeiras, que iniciaram a discussão sobre tal tema na década de 1970.

Era de se esperar que os processos das empresas brasileiras fossem concebidos a partir das experiências anteriores, cujos resultados já haviam sido publicados na época de sua criação. No entanto, em todas as empresas visitadas não foi encontrado nada semelhante ao controle de produtividade estabelecido pela Toshiba. Tal empresa mapeava, em 1978, o total de linhas de código em *Assembly* entregue por pessoa por mês. Em 1985, dez anos antes do estabelecimento das fábricas brasileiras, o percentual de reuso da Toshiba culminou em 48% (vide Tabela B.4). Já, 4 das 6 empresas visitadas não praticam, formalmente, a idéia de reuso de componentes. Em relação à qualidade do produto, a NEC e Fujitsu, possuem medidas acuradas, e não encontradas nos casos brasileiros. A NEC mapeava, na década de 1980, a quantidade de defeitos nas linhas de código de seus produtos (vide Tabela B.6). A Hitachi desenvolveu uma máquina de processo que funcionava de forma integrada a seu CASE com isto, dados como custo do projeto, tempo de trabalho despendido para execução de uma tarefa e o progresso dos envolvidos com o projeto eram capturados, automaticamente, possibilitando assim um ganho de produtividade. Uma ferramenta semelhante foi verificada somente na EMPRESA D.

Por fim, a superação das empresas brasileiras em relação às estrangeiras se verifica sobre uma perspectiva pontual: A atividade de equalização é bem formalizada nas empresas brasileiras, isto já não ocorre nas empresas estrangeiras.

11.2 Contribuições do Trabalho junto ao Pilar Empresarial

No pilar empresarial, o modelo foi aplicado por uma organização do setor produtivo de software, caracterizada como fábrica, na definição de um processo de

produção de uma de suas parceiras. Algumas contribuições podem ser verificadas com tal aplicação:

- Estabelecimento de uma fábrica de software, caracterizada com o ciclo curto de produção.
- Treinamento dos envolvidos com o processo, visto que quatro alunos foram recrutados, via contrato de estágio, para trabalhar no ambiente estabelecido pelo modelo e, atualmente, encontram-se trabalhando, definitivamente, em tal empresa.

Além das duas contribuições, este trabalho deixa um legado junto à instituição que se propôs a utilizar o modelo. Este fato pode ser comprovado, ao se efetuar uma leitura no capítulo 10, no qual é perceptível a criação do processo e da infraestrutura. Ambos estão disponíveis para receberem novos “residentes”, no próximo período letivo. O termo “residente”, até então não utilizado neste trabalho, é adaptado da área de medicina, no qual o aluno antes de atingir o mercado do trabalho, vivencia em um ambiente real (em nosso caso a fábrica) os aspectos práticos inerentes a sua formação, fomentando assim, a convergência entre os pilares acadêmico e empresarial.

Por fim, idéia da residência em fábrica de software vem ao encontro da necessidade de se enfatizar os aspectos da qualidade e produtividade nas disciplinas do núcleo de engenharia de software, fato este, também, destacado no capítulo introdutório deste trabalho.

11.3 Verificação das Proposições e a Resposta da Questão de Pesquisa

A possibilidade de se construir um mecanismo ou modelo que possibilite a criação ou a organização de um processo de produção para uma fábrica de software, deu a tônica para o desenvolvimento deste trabalho.

Ao analisar o modelo proposto no capítulo 5 é verificado, parcialmente, sob a luz da pesquisa de campo e dos relatos de CUSUMANO (1991) e, plenamente, por meio do experimento prático, que é possível concluir que o modelo pode auxiliar, “sim”, o engenheiro de processo a arquitetar um processo de produção de software.

Porém o “sim” destacado no parágrafo anterior exige que algumas considerações sejam efetuadas em relação aos modelos da qualidade que, também, auxiliam o engenheiro a organizar um processo de produção de software. Entre tais modelos se encontra o CMMI. Segundo FIORINI (2001), o CMMI pode ser utilizado para a definição de novos processos de produção de software. Neste trabalho, o CMMI é interpretado de uma outra maneira; segundo FABRI et. al. (2007), a proposta efetuada absorve o modelo em questão em sua estrutura interna e ele é utilizado para revestir as atividades e tarefas definidas nos ciclos de produção: longo, médio e curto. Os autores deixam claro que a criação e/ou organização do processo é de responsabilidade exclusiva das teorias de meta-processo e de reuso de processo, sendo que os modelos da qualidade são definidos como guias e estrutura de apoio. Esta visão é compartilhada pelo próprio CMMI, no qual este é caracterizado como um guia utilizado na melhoria do processo de produção de software, sendo que, aspectos de gerenciamento e manutenção de produtos e serviços não são contemplados. O CMMI não pode ser caracterizado como um processo devido à falta de algumas características como: definição de domínios de aplicações, modelagem tecnológica do processo e; modelos configuração de ativos (ou *templates*).

Além da questão de pesquisa, o capítulo 6 apresenta 3 proposições que devem ser verificadas sob a luz dos valores verdade (V - verdadeiro ou F - falso).

1. O processo de produção de uma fábrica de software é definido por meio de uma estrutura formal (métodos, procedimentos que levem à estruturação do processo de produção, algo semelhante às teorias de meta-processo e reuso de processo de software).
2. Existe um processo genérico de produção de software que pode ser aplicado ao contexto fabril.
3. O ciclo de produção e operação de uma fábrica de software é classificado como curto, médio e/ou longo.

Com base nos casos apresentados, é possível perceber que o processo de produção das fábricas é definido com base nas experiências dos profissionais que nelas atuam. Apenas a Toshiba e a Fujitsu utilizaram algum tipo de estratégia, não formalizada, além de tal experiência. Ressalta-se que 5 empresas brasileiras utilizaram o CMMI como um guia para a melhoria de seus processos. Com base nestas considerações é possível classificar a proposição 1 como “falsa”, uma vez que falta ao CMMI alguns parâmetros para que ele seja classificado como um meta-processo.

Já a proposição 2 pode ser classificada como verdadeira. Existe, sim, um processo de produção genérico que pode ser aplicado a um contexto fabril. Com base nos casos, é possível afirmar que o processo de produção de uma fábrica de software deve possuir as seguintes atividades (a serem refinadas):

- Gestão de projetos, da qualidade e de configuração;
- Modelagem de negócio (ou análise de sistemas);
- Projeto de software;

- Equalização do projeto de software;
- Codificação;
- Testes (unitário, integrado e de aceitação);
- Entrega do produto;
- Manutenção.

Analisando as atividades citadas, é possível verificar que o processo de produção das empresas é caracterizado como longo. Porém, as empresas brasileiras denominam como fábrica de software suas unidades organizacionais que realizam as atividades de equalização, codificação e teste (unitário e integrado) (vide capítulo 7 e anexo A). O fato citado neste parágrafo atribui um caráter de verdade para terceira proposição. Uma unidade fabril é caracterizada pelo ciclo curto, já a empresa que aglutina tal unidade pode ser caracterizada sob a luz dos três ciclos.

11.4 A Relação do Modelo com a Teoria de Fábrica de Software

A inserção do relacionamento do modelo proposto com a teoria de fábrica de software se faz necessário, pois a proposta do trabalho é, justamente, possibilitar a criação ou organização de um processo fabril de produção de software.

É possível verificar a relação entre a proposta e os modelos fabris delineados por CANTONE (1992) e LI et. al. (2001).

O modelo orientado à qualidade proposto por CANTONE (1992), salienta que uma estrutura fabril para a produção de software deve:

- possuir um modelo de processo;

- ser orientada a produto;
- definir um modelo da qualidade para a melhoria do processo de software.

Os itens acima foram verificados na definição de requisitos do modelo.

LI et. al. (2001), também, propõe um modelo orientado à qualidade focado nas normas ISO e CMM. Já o modelo proposto é mais abrangente, pois o mesmo está aberto a qualquer tipo de modelo da qualidade. Para os autores uma fábrica de software deve possuir:

- Aparato tecnológico definido (entidade 1): atributo verificado na definição de requisitos do modelo;
- Processo de produção definido e institucionalizado (entidade 2): Esta é a tônica do modelo, definir ou organizar processos;
- Definição dos envolvidos (entidade 3): A definição dos atores e envolvidos no processo de produção de software, também, faz parte da definição de requisitos do modelo;
- Definição da estrutura gerencial da fábrica (entidade 4): Esta entidade não é encontrada no modelo, pois o objetivo do mesmo é mapear as atividades da visão processual e não da visão gerencial;
- Ativos de processo e base de componentes (entidade 5): A definição dos ativos do processo é verificada na atividade de instanciação do modelo. Já o armazenamento dos componentes é direcionado pela máquina de processo.

11.5 Trabalhos Futuros

Alguns pontos, ainda, não foram tocados pelo presente trabalho, entre eles destacam-se:

- As atividades de seleção e adaptação de um processo não foram executadas, visto que apenas um processo para um projeto foi definido;
- Aplicar o modelo na definição de um processo de software caracterizado como ágil, e verificar o seu comportamento;
- Verificar o comportamento do modelo unido a outros guias utilizados na melhoria dos processos de software, como por exemplo: a norma ISO 12207;
- Verificar se o modelo é capaz de organizar um processo de produção de software caracterizado como distribuído, e;
- Aplicar o modelo na definição de novos processos, a fim de inferir novas evoluções para ele. A Fundação Educacional do Município de Assis mostra grande interesse que este ponto seja desenvolvido dentro de sua estrutura organizacional;
- Verificar a aderência do modelo aos requisitos essenciais para a sua concepção (veja seção 4.4). Esta aderência não foi contemplada neste trabalho, pois o modelo proposto foi aplicado somente na definição de um único processo, e não se encontra, totalmente, estabilizado.

Os pontos acima citados são de extrema importância e devem ser atacados em pesquisas subseqüentes a este trabalho.

Referências Bibliográficas

AMBLER, S. W. **Process Patterns Building Large-Scale Systems Using Object Technology**. Cambridge University Press - SIGS Books, Julho de 1998.

AVRILLIONIS, D.; CONRADI, R.; CUNIN, P-Y.; NGUYEN I. R.; Meta-Process. In: DERNIAME, Jean-Claude; KABA, Badara A.; WASTELL, David. **Software Process: Principles, Methodology, and Technology**. Series: Lecture Notes in Computer Science, Vol. 1500. 1999.

BASILI, V. R.; CALDIERA, G.; CANTOE, G.; A Reference Architecture for the Component Factory. **ACM Transaction on Software Engineering and Methodology**. v. 1. n. 1. p. 53-80. January 1992.

BECKER, U.; HAMANN, D.; VERLAGE, M. Descriptive Modeling of Software Processes, **IESIReport No. 045.97/E**, Version 1, Fraunhofer IESI, Dec. 31, 1997. Disponível para acesso em http://www.iese.fraunhofer.de/pdf_files/iese-045_97.pdf. Acessando em junho de 2006.

BEGOSSO, L. R.; FILGUEIRAS, L. V. L. Resultados da Implantação do Ambiente de Desenvolvimento de Maturidade em Engenharia de Software. In: **Anais do XI Workshop de Educação em Computação**. Campinas. 02 a 08 de agosto de 2003.

BEMER, R. W. The Economics of Program Production. In: **Information Processing**. Amsterdam: North-Holland Publ. Co.: v. II. n. 68. 1969.

BOEHM, B. A Spiral Model for Software Development and Enhancement. In: **IEEE Computer**. v. 21. n. 05. p. 61-72. Maio de 1988.

BURKHARDT, J.; HENN, H.; HEPPEL, S.; RINDTORFF, K.; SCHAECK, T. **Pervasive Computing – Technology and Architecture of Mobile Internet Application**. Addison Wesley. 2002.

CLEMENTS, P.; NORTHROP, L.. **Software Product Line: Practices and Partterns**. Boston. Addison Wesley. 2002.

CMMI-SE/SW (2002). **Capability Maturity Model Integration for System Engineering and Software Engineering CMMI-SE/SW, V1.2**. CMU/SEI-2002-TR-001 ESC-TR-2002-002. disponível em <http://www.sei.cmu.edu/cmmi/>.

CMMI-SE/SW (2002a). **Capability Maturity Model Integration for System Engineering and Software Engineering CMMI-SE/SW, V1.2**. CMU/SEI-2002-TR-002 ESC-TR-2002-002. Disponível em <http://www.sei.cmu.edu/cmmi/>.

CONRADI, R.; JACCHERI, M. L.; **Process Modeling Language**. In: DERNIAME, Jean-Claude; KABA, Badara A.; WASTELL, David. **Software Process: Principles, Methodology, and Technology**. Series: Lecture Notes in Computer Science, Vol. 1500. 1999.

CANTONE, Giovanni.; **Software Factory: Modeling the Improvement**. In: **Proceedings of the 3rd International Conference on Factory 2000**, Inglaterra: IEEE, 1992.

CORRÊA. L. H. **Teoria Geral da Administração: Uma Abordagem Histórica da Gestão de Produção e Operações**. São Paulo: Atlas, 2003.

CASSARO, Antônio C.; **Sistemas de Informações para Tomadas de Decisões – 2^a edição**. Pioneira, 1994.

COSTA, Ivanir. **Contribuição para o Aumento da Qualidade e Produtividade de uma Fábrica de Software através da Padronização do Processo de Recebimento de**

Serviços de Construção de Softwares - 174 pag.; Tese (Doutorado) Apresentada ao Departamento de Engenharia de Produção da Universidade de São Paulo; São Paulo. 2003.

CUSUMANO, M A. Software Factory: A Historical Interpretation. **IEEE Software**, v. 6. n. 2. p. 23-30. 1989.

CUSUMANO. M. A. **Japan's Softwares Factories**. New York: Oxford University Press. 1991.

CUSUMANO M. A.; SELBY R. W. How Microsoft Builds Software. In: **ACM Communication**. n. 40, p. 53-51. doi=<http://doi.acm.org/101145/25556.255698>. June, 1997.

DERNIAME, Jean-Claude; KABA, Badara A.; WASTELL, David. The Software Process: Modelling and Technology. In: _____. **Software Process: Principles, Methodology, and Technology**. Series: Lecture Notes in Computer Science, Vol. 1500. 1999.

DOUCET, F.; SHUKLA, S.; GUPTA, R.; OTSUKA, M. An Environment for Dynamic Component Composition for Efficient Co-Design; In: **Proceedings of the Conference on Design, Automation and Test in Europe**; Washington, DC. IEEE Computer Society, March, 04 - 08, 2002.

FABRI, J. A.; SPINOLA, M. M.; PESSOA, M. S. P. Unified Modeling Language and Relationship Management Methodology in the Development of Applications for Distance Teach. In: **Resources Management Association Conference. Information Technology and Organizations: Trends, Issues, Challenges and Solutions**. Philadelphia USA: Idea Group Publishing, 2003.

FABRI, J. A.; L'ERÁRIO, A.; TRINDADE, A. L. P.; PESSOA, M. S. de P.; SPÍNOLA, M. M. Desenvolvimento de um método de Configuração e Replicação de Fábrica de

Software – tutorial apresentado. In: **4ª Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento**. Madrid - Espanha. Novembro de 2004.

FABRI, José A.; TRINDADE, André L. P.; L'ERÁRIO, Alexandre; PESSÔA, Marcelo S. P.; SPINOLA, Mauro M.; Desenvolvimento e Replicação de uma Fábrica de Software – tutorial. In: **Anais do VI SIMPROS – Simpósio Internacional de Melhoria de Processos de Software**. Brasil: SENAC/CenPRA, novembro de 2004(a).

FABRI, J. A.; TRINDADE, A. L. P.; BEGOSSO, L. R.; L'ERÁRIO, A.; SILVEIRA, F. L. F.; PESSÔA M. S. de P. Techniques for the Development of a Software Factory: Case CEPEIN-FEMA. In: **ICSSEA International Conference Software & Systems Engineering and their Applications**. 17th. November 30 - December 3, Paris, 2004(b).

FABRI, J. A.; TRINDADE A. L. P.; DURSCKI, R.; PESSOA, M. S. de P.; SPINOLA M. Proposta de um Mecanismo de Desenvolvimento e Customização de uma Fábrica de Software Orientada a Domínios. In: **Conferência Latino-Americana de Informática**. XXXI. Santiago de Cali Colômbia. 10 a 14 de outubro de 2005.

FABRI, J. A.; TRINDADE A. L. P.; OLIVEIRA. A. C. M. T. G.; PESSOA, M. S. de P.; OLIVEIRA J. C. G. A Importância da Abordagem dos Conceitos de Metodologia de Pesquisa para os Cursos de Ciência da Computação. In: **XIII Congreso Iberoamericano de Educación Superior em Computación**. Santiago de Cali Colômbia. 10 a 14 de outubro de 2005(a).

FABRI, J. A.; BEGOSO, L. C.; BEGOSSO, L. R. Ensino de Engenharia se Software Utilizando um Ambiente Maduro Permeado pelo Conceito de Fábrica de Software In: **Procedings Global Congress on Engineering and Technology Education – IEEE**. Bertioga – SP – Brasil. 13 a 16 de março de 2005b.

FABRI, J. A.; BEGOSSO, L. R.; PESSÔA, M. S. de P.; SPÍNOLA, M. M. Desenvolvimento do Conceito de Fábrica de Software em Instituições de Ensino que possuem Cursos de Computação. In: **Revista ProQualiti – Qualidade na Produção de Software**. Vol. 2, n. 1, Lavras – Minas Gerais. 2006a.

FABRI, J. A.; TRINDADE, A. L. P.; CAMARGO, V. L. Recursos Humanos Necessários em uma Estrutura Fabril para o Desenvolvimento de Software. In: **I Workshop de Trabalhos de Pós-graduação da Faculdade de Tecnologia de São Paulo**. São Paulo. 18 e 19 de outubro de 2006b.

FABRI, J. A.; TRINDADE, A. L. P.; L'ERÁRIO, A.; SILVEIRA, M.; PESSOA, M. S. de P. O Papel do CMMI na Configuração de um Meta-Processo de Produção de Software com Características Fabris: Um Estudo de Caso. In: **Anais da 6ª Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento**. Lima, Peru. 2007. No prelo.

FABRI, J. A.; TRINDADE, A. L. P.; L'ERÁRIO, A.; PESSOA, M. S. A. A Organização da uma Máquina de Processo e a Melhoria do Processo de Produção de Software em um Ambiente de Fábrica. In: **Anais da 6ª Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento**. Lima, Peru. 2007a. No prelo.

FABRI, J. A.; SCHEIBLE, A. C. F.; MOREIRA, P. M. L., TRINDADE, A. L. P.; BEGOSSO, L. R.; BRAUN A. P.; PESSÔA, M S. de P. Meta-Process used for Production Process Modeling of a Software Factory: the Unitech Case. In: **Managing Worldwide Operations and Communications with Information Technology - IRMA 2007 Proceedings**. Vancouver, Canadá. 2007b. No prelo.

FABRI, J. A.; TRINDADE, A. L. P.; BEGOSSO, L. R.; PESSOA, M. S.; L'ERÁRIO, A. The Use of the IDEF-0 to Model the Process in a Software Factory. In: **Managing Worldwide Operations and Communications with Information Technology - IRMA 2007 Proceedings**. Vancouver, Canadá. 2007c. No prelo.

FABRI, J. A.; PESSOA, M. S. Como Organizar um Processo Fabril de Produção de Software. In: **Anais da 6ª Jornada Ibero-Americana de Engenharia de Software e Engenharia de Conhecimento**. Lima, Peru. 2007d. No prelo.

FAYAD, W. E.; SCHMIDT, D. C.; JOHNSON, R. E. **Building Application Framework**, ISBN 0471-24875-4. 1999.

FELICIANO NETO, Acácio; SHIMIZU, Tamio; **Sistemas Flexíveis de Informações**; Brasil: Makron books, 1996.

FERNANDES, A. A. **O Paradigma da Fábrica de Software Itens Qualificadores e Ganhadores de Pedidos e Práticas das Empresas de Informática do Brasil**. Tese de doutorado apresentada ao Departamento de Engenharia da Produção da Escola Politécnica da Universidade de São Paulo. 476 p. 2000.

FERNANDES, A. A. TEIXEIRA, D. de S. **Fábrica de Software: Implementação e Gestão de Operações**. Editora Atlas. 2004.

FERNSTROM, C. The Eureka Software Factory: Concepts and Accomplishment. In: **Proceedings Third European Software Engineering Conference**. Berlin, Alemanha. 1991.

FERNSTROM, C. NARFELT K. H. OHLSSON L. Software Factory Principles, Architecture and Experiments. **IEEE Software**. v. 9. n 2. p. 36-44. March and April 1992.

FIORINI, Soeli Teresinha. **Arquitetura para Reutilização de Processos de Software**. Tese de Doutorado Apresentada ao Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro. 2001.

FRANCH, X.; RIBÓ, J. **Supporting Process Reuse in PROMENADE**. Departament de Llenguages i Sistemes Informàtics, Universitat Politècnica de Catalunya – Barcelona 2002. (Research Report, No. LSI-02-14-R). Disponível em: http://www.lsi.upc.edu/dept/techreps/llicitat_detallat.php?id=571. Acesso em maio. 2004.

GATLIN, JONATHAN. **Bill Gates - The Path To The Future**. Ed. Avon Books 1ª Edição. 1999

GOLOMSHTOK, A. **.NET System Management Services**. Ed. APRESS. 1ª Ed. 2003.

GOMAA, H.; KERSCHBERG, L.; FARRUKH, G. A. Domain Modeling of Software Process Models. In: **Proceedings of the Sixth IEEE International Conference on Engineering of Complex Computer Systems**. p. 50-61. 2000.

GRANVILLE, M.; DAVE, A. **Extreme Programming – Guia Prático**. Editora Campus. 1a. Edição. 2002.

HOLLENBACH, C., FRAKES, W.: Software Process Reuse in an Industrial Setting. In: **Fourth International Conference on Software Reuse**, Orlando, Florida, IEEE Computer Society Press, Los Alamitos, CA, 1996.

HUMPHREY, Watts S. **Managing the Software Process**. 1ª Ed. Addison Wesley, 1989.

HUMPHREY, Watts S.; Software and the Factory Paradigm; In: **Software Engineering Journal**. EUA: IEEE, setembro de 1991.

HUMPHREY, Watts S. **TSP - Leading a Development Team**. Coleção: SEI SERIES IN SOFTWARE ENGINEERING. Editora: ADDISON WESLEY. 2005.

HUMPHREY, Watts S. **PSP(SM) A Self-Improvement Process for Software Engineers**. Editora: ADDISON WESLEY. 2005a.

HENNINGER Scott. An Environment for Reusing Software Processes. In: **Appears in International Conference on Software Reuse**. pp. 103-112, 1998.

IDEF0. **Draft Federal Information Processing Standards Publication 183** 1993 December 21 Announcing the Standard for INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0).

IEEE Std 830-1998. **Recommended Practice for Software Requirements Specifications** IEEE Computer Society. Approved 25 June 1998.

JÄGER, D.; SCHLEICHER, A.; WESTFECHTEL, B. Using UML for Software Process Modeling. In: **Proceedings of the 7th European Software Engineering Conference. 7th ACM SIGSOFT International Symposium on Foundations of Software Engineering**. Toulouse, France, September 06 - 10, 1999.

JØRGENSEN, H.; CARLSEN, S. Writings in Process Knowledge Management: Management of Knowledge Captured by Process Models. In: **SINTEF Telecom and Informatics, 2001**. Oslo (Technical Report, No. STF40A00011). Disponível em: <http://www.informatics.sintef.no/~hdj/>. Acesso em dezembro. 2003.

KOLSTE, B.; PETERSEN, D. **Oracle Power Objects Handbook**. Ed. OSBORNE - MCGRAW-HIL. 1ª. Edição. 1995.

KRUCHTEN, P. **Introdução ao RUP - Rational Unified Process**. Editora Ciência Moderna. 1ª Edição. 2003.

KWASNICKA, Eunice L.; **Teoria Geral da Administração: uma síntese – 2ª edição**; Brasil: Atlas, 1989.

L'ERÁRIO, A.; PRADO, T. F.; SPÍNOLA, M. M.; PESSÔA, M. S. de P.; FABRI, J. A. Desenvolvimento Distribuído de Software para Sistemas Pervasivos: Um Estudo de Caso. In: I Anais do Simpósio Brasileiro de Sistemas de Informação. Porto Alegre - RS. p. 163 a 170. 2004.

L'ERÁRIO Alexandre. **Processo Distribuído para o Desenvolvimento de Software**. Trabalho de Qualificação de Doutorado Apresentado ao Departamento de Engenharia de Produção da Escola Politécnica da Universidade de São Paulo. Julho de 2006.

LARMAN, C. Utilizando UML e Padrões: **Uma Introdução a Análise e ao Projeto Orientados a Objetos**. 3ª. Edição. Bookman 2000.

LOY, M.; ECKSTEIN, R.; ELLIOTT, J. **Java Swing**. 2ª Ed. OREILLY & ASSOC. 2002.

LI, C.; LI, H.; LI, M. A Software Factory Model Based on ISO 9000 e CMM for Chinese Small Organization. In: **Asia-Pacific Conference on Quality Software (APAQs'01) Proceeding**. Hong Kong. December, 2001.

MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. Editora Livros Técnicos Científicos. 3ª. Edição. 2002.

MELO, Ivo S.; **Administração de Sistemas de Informação**; Brasil: Pioneira, 1999.

NADLER, David A.; TUSHMAN, Michael L. **Competing by Design - The Power of Organizational Architecture**. Oxford University Press. 1999.

NAKANO, D.; FLEURY, A. Métodos de Pesquisa em Engenharia de Produção. In: **Anais do Encontro Nacional de Engenharia de Produção**. Piracicaba 1996.

NATSUI, Érica; **Inteligência Competitiva**; trabalho de conclusão de curso apresentado à Faculdade de Economia, Administração e Contabilidade da USP; Brasil: FEA-USP, 2002.

NELSON, M. L. **Barriers to Software Reuse and the Projected Impact of World Wide Web on Software Reuse**. Acessado a partir do endereço <http://citeseer.ist.psu.edu/nelson96barriers.html> em julho de 2006. Ano de publicação: 1996.

PAULA FILHO, W. P. **Engenharia de Software: Métodos Fundamentos Métodos e Padrões**. Rio: LTC. 2ª ed. 2003.

PETERS, J. F.; PEDRTOZ, W. **Engenharia de Software: Teoria e Prática**. Rio: Ed. Campus 2001.

PERRY, D. E. Practical Issues in Process Reuse, In: **10th International Software Process Workshop**. França, Junho de 1996.

PERRY, D. E. Using Process Modeling for Process Understanding. In: **Software Process Improvement**. Espanha. Barcelona. 1997.

PRIETO-DÍAZ, R. Implementing faceted classification for software reuse. In: **Communication ACM**. vol. 34, n. 5 p. 88-97. Maio de 1991. Disponível para acesso em <http://doi.acm.org/10.1145/103167.103176>. Acessado em 2006.

PRESSMAN, R. S. **Engenharia de Software**. Rio de Janeiro: McGraw-Hill, 2002.

PRATES, R. **Clipper 5.2: Guia De Consulta Rápida**. São Paulo: Novatec, 1996.

REIS, R. Q. **APSEE: Um Meta-Modelo para Apoiar a Reutilização de Processo de Software**. Tese de doutorado apresentada ao Instituto de Informática da Universidade Federal do Rio Grande do Sul. 2002.

ROCHA T. A.; OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. Adequação de Processos para Fábrica de Software. In: **Anais do VI Simpósio Internacional de Melhoria de Processo de Software**. São Paulo, 24 a 26 de novembro de 2004.

ROCKWELL, R.; GERA, M. H. The Eureka Software Factory CoRe: A Conceptual Reference Model for Software Factories. In: **Software Engineering Environments Conference Proceedings**. P. 80 – 93. July 1993.

RUHE, G.; Experience Factory-Based Professional Education and Training; In: **Proceedings of the 12th Conference on Software Engineering Education and Training**; Alemanha: IEEE, 1999.

SEAMAN, C. B.; AAA: A modeling Language for Software production Environments. In: **Proceedings of the 1992 Conference of the Centre For Advanced Studies on Collaborative Research**. Vol. 1. J. Botsford, A. Ryman, J. Slonim, and D. Taylor, Eds. IBM Centre for Advanced Studies Conference. IBM Press, 513-530. 1992.

SCHAEFER, Wilhelm.; FUGGETTA, Alfonse.; GOBART Claude.; JAHNKE Jens. Architectural Views and Alternatives. In: DERNIAME, Jean-Claude; KABA, Badara A.; WASTELL, David. **Software Process: Principles, Methodology, and Technology**. Series: Lecture Notes in Computer Science, Vol. 1500. 1999.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da Produção**. São Paulo: 2ª Ed. Atlas, 2002.

SOMMERVILLE I. **Engenharia de Software**. 6ª Edição. Addison Wesley, 2003.

SOMMERVILLE I. **Software Engineering**. 8ª Edição. Addison Wesley, 2006.

SPEM. Software Process Engineering Metamodel Specification. Object Management Group. Janeiro 2005. Versão 1.1 Formal/05-01-06. Disponível para

acesso em www.omg.org/technology/documents/formal/spem.htm. Acessado em fevereiro de 2006.

SPÍNOLA, M. M.; PESSÔA, M. S. de P.; FABRI, J. A.; COSTA, I.; DIPPOLITO, E.; ElabTI: Um ambiente real e replicável de produção de software. In: **Programa Brasileiro da Qualidade e Produtividade em Software (PBQP)**. 3ª Edição, revisada e ampliada. Setembro de 2004.

SUCCI, G.; BENEDICENTI, L.; PREDONZANI, P.; VERNAZZA, T.; Standardizing the Reuse of Software Processes. **SatandardView Journal**. v. 5. n. 2. ACM PRESS New York. Junho de 1997.

TEDLOW, R. S. **Sete Homens e os Impérios que Construíram**. S. Paulo: Futura 2002.

THO. I. **Managing the Risk of IT Outsourcing**. Butterworth – Heineman. 1a. Edição. 2005.

THORESON, S. A.; The Automated software Development Project at McDonnell Aircraft Company (the software factory). In: **Proceedings of the National Aerospace and Electronics Conference**. EUA: IEEE, 1989.

TRINDADE, A. L. P. **Uma Contribuição para o Entendimento dos Papéis do Conhecimento e da Ensinagem em um Ambiente de Fábrica de Software**. Tese de Doutorado apresentado ao Programa de Doutorado em Engenharia de Produção da Universidade de São Paulo – Escola Politécnica da Universidade de São Paulo. 2006.

TRINDADE, A. L. P. **Métricas para Orçamento e Planejamento da Produção de Software**. Dissertação de mestrado apresentada ao Programa Pós-Graduação em Engenharia de Produção da Universidade de São Paulo – Escola Politécnica da Universidade de São Paulo. 1999.

VERRALL, Malcom S.; ESF: The software bus. In: **Colloquium on Architectures for Distributed Development Support Environments**. Inglaterra: IEEE, 1991.

WANG, J. A.; Towards component-based software engineering. In: **Proceedings of the Eighth Annual Consortium on Computing in Small Colleges Rocky Mountain Conference**. USA: Consortium for Computing Sciences in Colleges. 2000.

WELLING, L.; THOMPSON, L. **PHP e MYSQL Desenvolvimento Web**. 3ª. Edição. Editora Campus. 2005

WILES, E. **Economic models of software reuse: A survey, comparison and partial validation**. Acessado a partir do endereço <http://citeseer.ist.psu.edu/wiles99economic.html> em julho de 2006. Ano de publicação: 1999.

YACOUB, S.; AMMAR, H.; MILI, A. **Characterizing a Software Component**. <http://www.sei.cmu.edu/cbs/icse99/papers/34/34.htm>, acessado em julho de 2005.

YIN, Robert K. **Estudo de Caso: Planejamento e Método**. 3 ed. traduzida. Bookman. Porto Alegre. 2005.

ANEXO A – Relato Detalhado das Fábricas Brasileiras

Este anexo tem como objetivo, apresentar o relato detalhado de seis fábricas de software atuantes no território brasileiro. Cinco delas possuem certificações oficiais da qualidade relacionadas ao modelo CMMI e uma possui certificação oficial relacionada a norma ISO 9001-2000, fator este determinante na seleção de tais empresas para a composição deste trabalho. Salienta-se, novamente, que não existe uma autorização formal para divulgar o nome das empresas aqui descritas, com isto utilizou-se a denominação proposta na Tabela 6.2 para as mesmas.

Estruturalmente, este anexo está dividido em 6 seções, cada uma delas apresenta o caso de uma empresa. Cada seção foi dividida em 7 subseções, respeitando o roteiro para desenvolvimento do estudo de caso apresentado na tabela 6.3, denominadas de seguinte forma: caracterização da empresa, insumos do processo, atividades do processo, gestão operacional, armazenamento de dados, produtos gerados e forma de criação e organização do processo. É importante salientar que os termos e as figuras, utilizadas na descrição dos casos foram inferidos a partir das entrevistas realizadas, fato este que gerou uma ausência na padronização em relação a eles.

Por fim, os fatos semelhantes ou idênticos encontrados no processo de produção de empresas distintas são relatados somente uma única vez.

A.1 A Fábrica de Software da EMPRESA A

A.1.1 Caracterização da Empresa

A EMPRESA A atua na área de engenharia de software desde 1995 e tem como meta principal oferecer soluções em Tecnologia da Informação (TI) para os seguintes seguimentos: Gestão Empresarial - *Enterprise Resource Planning* (ERP); Gerenciamento de Relacionamento com o Cliente (*Customer Relationship*

Management (CRM)) e Administração Tributária. Tal empresa também presta serviços de consultoria, suporte, planejamento e desenvolvimento de soluções em TI para otimizar o processo produtivo das organizações. Atualmente, a empresa possui cerca de 1200 colaboradores¹ (nome empregado para designar o conceito de trabalhador ou funcionário) divididos entre a matriz, onde se localiza a fábrica de software; quatro filiais e em clientes espalhados por todo o território nacional.

A missão da empresa em questão é: Criar diferenciais competitivos para os clientes, consolidando as suas estratégias de negócios, por meio da aplicação da TI, respeitando os valores éticos, as normas legais organizacionais, atendendo aos requisitos dos clientes e promovendo a sua satisfação, pela melhoria contínua dos processos e da gestão. Oferecer às organizações públicas e privadas soluções viáveis de alta performance na área de TI, também, constitui um dos objetivos da empresa.

Atualmente, a empresa está estruturada em 3 grandes áreas:

- Consultoria: Foco em alinhamentos estratégico de TI e negócios, governança e gestão de TI;
- Desenvolvimento de software: Foco em desenvolvimento de aplicações sobre medida junto a clientes;
- *Outsourcing*²: Foco em gestão de sistemas: desenvolvimento manutenção e suporte, escritório de projetos e gestão de infra-estrutura.

Em relação a certificações da qualidade, a EMPRESA A possui ISO 9001:2000, ISO 14001:2004 e CMMI nível 3 obtida em abril de 2006³.

¹ Somente a fábrica de software possui cerca de 130 colaboradores.

² Segundo THO (2005), a idéia de outsourcing pode ser exemplificada da seguinte forma: Uma determinada empresa, localizada no país A, contrata outra empresa, localizada no país B para a produção de código fonte.

³ Em agosto de 2005, época em que o estudo de caso foi desenvolvido, a EMPRESA A possuía certificação CMMI nível 2.

A.1.2 Insumos do Processo

Nas entrevistas realizadas, foi possível detectar que a fábrica de software recebe, na maioria das vezes⁵, projetos de software para codificação e teste, caracterizando-a com o ciclo curto.

A modelagem de negócio e o projeto de software são desenvolvidos pelas filiais, denominadas clientes internos da fábrica, e pelos próprios clientes, denominados clientes externos. Isto leva a concluir que a EMPRESA A (não a fábrica) está configurada sobre a luz do ciclo longo. Uma representação ilustrativa, apresentando os insumos presentes no processo de produção da EMPRESA A, pode ser verificada por meio da Figura A.2.

Figura A.2 – Modelo de Produção da EMPRESA A

Na Figura A.2 é perceptível a presença de 3 visões, descritas nos itens abaixo:

⁵ O termo na “maioria das vezes” foi empregado pelo diretor de produção. Ele salienta, também, que a fábrica de software está preparada para desenvolver todo o ciclo do projeto, porém isso, raramente, é feito.

- Visão 1: cliente externo contrata uma determinada filial para a realização da modelagem de negócio. O cliente externo também pode contratar, diretamente, a fábrica de software (na qual esta é responsável pela codificação e teste). Neste caso ele deve possuir as competências para o desenvolvimento do modelo de negócio⁶ e do projeto de software. caso o cliente externo não possua tais competências, a fábrica de software estabelece uma equipe para a realização de tais atividades;
- Visão 2: cliente interno desenvolve o modelo de negócio e o projeto de software para o cliente externo, e contrata a fábrica de software para codificação e teste. O cliente interno também pode optar por desenvolver o ciclo projeto de software, incluindo a codificação e o teste, sem contratar a fábrica;
- Visão 3: realização da codificação e teste (ciclo curto) com base no projeto de software, tal projeto pode vir de uma filial, de uma equipe estabelecida pela fábrica de software na visão 1 ou, diretamente, do cliente externo.

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção.

A.1.3 Atividades do Processo

As atividades pertinentes ao processo de software da EMPRESA A são: levantamento e especificação dos requisitos; análise de sistemas (ou modelo de negócio); projeto de software; equalização do projeto de software; codificação; teste de software; homologação e implantação. Este relato apresenta somente uma descrição das atividades relacionadas ao ciclo curto, pois o estudo de caso foi realizado somente na unidade fabril da EMPRESA A.

⁶ Isto leva a fábrica de software a analisar os artefatos desenvolvidos pelos clientes. Somente se os documentos produzidos pelos clientes tiverem certo grau de qualidade são utilizados.

A Atividade de Equalização

A atividade de equalização tem como objetivo verificar a “corretude” e a “completude” das especificações advindas da atividade de projeto de software para que as mesmas possam ser colocadas em linha de produção (codificação e teste). Esta atividade está sob responsabilidade do equalizador.

Na atividade de equalização, um projeto de software é dividido em módulos. Os módulos, por sua vez, são divididos em serviços, nos quais estes possuem um conjunto de artefatos⁷. É sobre os artefatos, estabelecidos na ordem de serviço, que o equalizador verifica os fatores de “completude” e “corretude”.

Quando a ordem serviço não é entendida ou está inconsistente e incompleta, a pessoa responsável pela equalização entra em contato com a equipe de projeto de software (cliente interno ou externo da fábrica) para que os problemas sejam solucionados.

A equalização demanda o preenchimento de um relatório, fato este comprovado na Figura A.3. Em tal documento são descritos os problemas e o histórico⁸ relacionado à equalização de uma ordem de serviço.

É importante salientar que a atividade de equalização é uma atividade contínua, isto é, ela ocorre, paralelamente, a codificação da ordem de serviço. A Figura A.3 mostra o momento de início da atividade de equalização, na qual um maior número de problemas tendem a ser detectados e, com o passar do tempo, eles vão diminuindo e chegam a zero. A atividade de equalização é finalizada após a análise do relatório de equalização por parte da equipe de projeto.

⁷ Alguns exemplos de artefatos de projeto: Modelo de dados, protótipos, diagramas - estados, seqüência, classes (importante: o número de diagramas pode variar de projeto para projeto). A linguagem utilizada na Fábrica de Software é UML, e existe uma granularidade comum para o desenvolvimento dos diagramas.

⁸ O histórico da equalização é concretizado, totalmente, somente quando o(s) problema(s) é(são) resolvido(s).

Figura A.3 – A atividade de equalização – **EMPRESA A**

Os artefatos para o desenvolvimento da atividade em questão são:

- *checklist* para equalização: Apresenta, ao equalizador, o que necessita ser equalizado;
- Documento de rejeite da ordem de serviço: Utilizado para informar à equipe de projeto que a equalização foi rejeitada. Uma equalização é rejeitada somente quando a ordem de serviço não atende a uma pontuação mínima;
- Guia de pontuação: Documento utilizado para pontuar uma ordem de serviço, fatores de “completude” e “corretude” são utilizados em tal guia;
- Relatório de equalização: Artefato aglutinador, cujo objetivo é consolidar todas as informações geradas pela atividade em questão. A **EMPRESA A** caracteriza tal relatório como histórico da equalização.

Por fim, o equalizador deve realizar o apontamento de horas trabalhadas na máquina de processo. Tal apontamento será utilizado pelo líder de equipe e pelo gerente da fábrica de software para o acompanhamento do projeto.

A Atividade de Codificação

A atividade de codificação é desenvolvida sob a ótica do desenvolvimento multicamadas⁹. No momento da codificação, o desenvolvedor recebe os artefatos gerados pela atividade de equalização e as especificações pertinentes à ordem de serviço. Ao recebê-las ele verifica se não existem erros ou inconsistências (não percebidos na atividade, anteriormente, descrita) e, caso algum problema seja detectado, uma nova equalização deve ser desenvolvida. A solicitação desta deve ser feita, formalmente, pelo líder de equipe¹⁰. Tal solicitação é inserida no histórico de equalização e será utilizada na composição dos fatores da qualidade, descritos mais adiante.

Um fato importante citado pelos envolvidos com a fábrica de software da EMPRESA A, é a existência de uma plataforma de desenvolvimento¹¹, operando nas tecnologias JAVA (LOY et. al. (2002)) e .NET (GOLOMSHTOK (2003)). A plataforma utilizada é comercializável junto ao mercado, caracterizando não somente a venda de software por parte da fábrica, mas também a de uma estrutura componentizada, a qual pode ser utilizada por outras empresas que desenvolvem software.

A plataforma de desenvolvimento funciona como uma espécie de biblioteca de componentes, os desenvolvedores encaixam alguns trechos de código (vide Figura A.4) nas classes instanciadas para o desenvolvimento das funcionalidades espelhadas nas ordens de serviço.

⁹ No desenvolvimento multicamadas, o programador separa os procedimentos que mapeiam um programa em camada de interface, camada de regras e camada de dados.

¹⁰ Salienta-se que o desenvolvedor discute com o líder de equipe a necessidade da realização de uma nova equalização para uma determinada ordem de serviço.

¹¹ A plataforma de desenvolvimento foi concebida sob a ótica da orientação a objetos. Nela existe um conjunto de classes, que são instanciadas pelos desenvolvedores, na qual os métodos são recheados com o código específico relacionado ao produto em desenvolvimento (vide Figura A.4).

Figura A.4 – Instanciando código a partir da plataforma de desenvolvimento – **EMPRESA A**

Outro aspecto interessante, detectado no caso, é a presença do teste unitário¹² na atividade de codificação. É perceptível que todo desenvolvedor é obrigado a desenvolver tal teste. Um *checklist* apóia a realização de tal tarefa.

Além da codificação e teste, o desenvolvedor deve realizar o apontamento de horas trabalhadas na máquina de processo. Tal apontamento será utilizado pelo líder de equipe e pelo gerente da fábrica de software, para o acompanhamento do projeto.

Por fim, a atividade de codificação utiliza a plataforma e o padrão para desenvolvimento de código, juntamente com o *checklist* de teste como artefatos.

Teste e Entrega

A atividade de teste tem como objetivo efetuar o teste de caixa preta. Para o desenvolvimento de tal atividade, a fábrica de software em questão possui uma célula especial, denominada célula de teste. O líder da célula de teste relata que o desenvolvedor é responsável pelos testes de primeiro nível (teste unitário). Uma

¹² Dentro do processo de produção, a **EMPRESA A** utiliza o termo teste de primeiro nível para contextualizar o conceito de teste unitário.

informação interessante, presente na fábrica de software da EMPRESA A, é o fato da existência de vários *checklists* para a atividade de teste. Cada *checklist* é aplicado a um tipo de componente ou a uma unidade de software, por exemplo: existem *checklists* utilizados em unidades de software (ou componentes) caracterizadas como relatórios e outros aplicados em componentes utilizados na inserção de dados. É de responsabilidade da célula de teste prover o *checklist* específico para cada tipo de teste.

No momento que o desenvolvedor realiza o teste unitário de primeiro nível, o líder de equipe gera uma versão e disponibiliza-a para a célula de teste. A partir daí, tal célula entra em operação. Na execução da atividade, a célula em questão realiza um outro teste unitário, denominado pela fábrica de software como teste unitário de segundo nível. Após o desenvolvimento desta tarefa, é realizado o teste de integração de componentes de uma ordem de serviço (teste de integração de primeiro nível). Posteriormente, as ordens de serviços são integradas e o teste de integração de ordens de serviços é desenvolvido (teste de integração de segundo nível (vide Figura A.5)).

Atualmente, a atividade em questão prevê que os casos de teste devem ser disponibilizados pela equipe de projeto de software (cliente interno), porém esta prática não é, totalmente, realizada por ela. Quando os casos de testes não são enviados, os mesmos são desenvolvidos pela célula de teste. O coordenador da célula de teste também relata que a EMPRESA A possui um banco de teste, que é utilizado quando os casos de testes não são enviados pela equipe de projeto de software (ou cliente interno).

Quando o teste não encontra nenhum problema relacionado ao produto, o mesmo pode ser entregue para o cliente (interno ou externo).

Cabe ressaltar que todos os testadores interagem, diretamente, com a máquina de processo, informando o tempo que o componente e a ordem de serviço que ficaram sob responsabilidade de tal célula, a quantidade e a natureza dos erros

encontrados também são informados. Estas informações são utilizadas, posteriormente, para o gerenciamento e controle da qualidade da fábrica de software.

Figura A.5 – Procedimento de teste desenvolvido pela fábrica de software da EMPRESA A.

Após a apresentação das atividades operacionais de fábrica, e seguindo o roteiro estabelecido na Tabela 6.3, as próximas informações a serem descritas neste texto relacionam-se com modelo de processo utilizado pela fábrica software em questão.

O modelo de processo de software que norteia a produção da EMPRESA A é o incremental (vide definição de tal modelo no capítulo 4). Este fato é perceptível ao analisar a figura do Analista de Sistema, cuja responsabilidade é colher os requisitos do produto junto ao cliente e desenvolver o modelo de negócio. Tal modelo é projetado na forma de um software. O projeto de software é “fatiado” em ordens de serviços, e estas são codificadas, testadas, homologadas e implantadas, gradualmente, caracterizando assim o aspecto de “incrementabilidade” do processo.

Uma outra constatação detectada é a participação dos envolvidos na execução das atividades do processo. Na descrição das atividades processuais foi possível verificar a presença da equipe de projeto de software (EP), do equalizador (E), do desenvolvedor (D) e do testador (T). Além destes atores, existem outros que fazem parte do processo de produção de software, alguns deles se relacionam com a visão processual e outros se relacionam com a visão gerencial¹³. Os atores que participam, ativamente, da organização em questão são:

- Analista de Sistemas (AS): O analista de sistemas é responsável por colher os requisitos sistêmicos e de software. O analista de sistemas, também, é responsável pela modelagem do negócio da organização cliente da empresa em questão;
- Coordenador da Célula de Testes (CT) – é um papel gerencial que tem a responsabilidade de apoiar a realização dos testes funcionais;
- Coordenador da Produção (CP) – é um papel gerencial dentro da equipe, com as responsabilidades de: coordenar as equipes de equalização, implementação e testes; planejar e acompanhar projetos; orientar, tecnicamente, a construção dos softwares garantindo a utilização das metodologias, procedimentos, padrões e boas práticas na utilização das tecnologias adotadas; e assessorar a gerência da fábrica. Este papel demanda conhecimentos de gerência de configuração e de projetos;
- Coordenador do Controle da Qualidade (CCQ) – é um papel gerencial que tem a responsabilidade de planejar inspeções nos software em produção, coordenar e aprovar a definição de padrões de nomenclatura e interface na Fábrica, *templates* (padrões) de implementação, boas práticas de programação e base de conhecimento de problemas;

¹³ As visões processual e gerencial de uma fábrica de software foram apresentadas no capítulo 2 (vide Figura 2.3).

- Desenvolvedor (D) - é um papel operacional e direcionado para o domínio de técnicas de programação e ferramentas de desenvolvimento, com a responsabilidade de codificar e testar as unidades de software (componentes, formulários, relatórios) que forem especificadas para produção. Os desenvolvedores poderão atuar na implementação destas unidades de implementação e no desenvolvimento dos testes;
- Diretor de Produção de Software (DP) – é um papel gerencial de nível sênior, com a responsabilidade sobre todo o processo (níveis: tático e operacional) da Fábrica de Software;
- Equalizador (E) – é um papel operacional e direcionado para o domínio de técnicas de modelagem e metodologias de desenvolvimento de software, com a responsabilidade de equalizar a especificação fornecida pelo cliente apontando dúvidas, falhas e inconsistências e auxiliando, assim, a equipe de desenvolvedores no entendimento da especificação;
- Equipe de Especificação (EE), também chamada de Equipe de Projeto (EP) – será responsável pela definição e organização de requisitos, modelagem e projeto físico do software e fornecimento destas informações para a Fábrica de Software, assim, como esclarecer dúvidas surgidas ao longo do processo de produção e ajustar a especificação quando necessário. Também caberá à equipe de especificação a validação dos produtos entregues. Esta equipe é composta na maioria das vezes por analistas de sistemas.
- Garantia da Qualidade de Software (Software Quality Assurance - SQA) – é um papel que tem a responsabilidade de verificar a conformidade das atividades e produtos de software de todos os projetos, de acordo com os procedimentos definidos na Fábrica de Software;

- Gerente da Fábrica de Software (GFS) – é um papel gerencial dentro da equipe, com a responsabilidade de avaliar novos serviços, elaborar propostas técnicas e acompanhar a execução de todos os projetos em andamento, de forma que cada software esteja de acordo com a especificação informada e que sejam produzidos com qualidade e nos prazos e orçamento estabelecidos. Esta pessoa, também, é responsável pela prospecção de novas tecnologias e metodologias de trabalho para a Fábrica de Software;
- Grupo de Processo e Engenharia de Software (SEPG) - é um grupo formado pelo coordenador de produção, gerente da fábrica de software e diretor de produção, alguns líderes de equipe e SQA. Tal grupo tem como objetivo definir, acompanhar e avaliar o processo de software;
- Líder de Equipe (LE) – é um papel operacional dentro da equipe, com a responsabilidade de assessorar a coordenação de produção em atividades de acompanhamento de projetos e orientação técnica da equipe de produção;
- Técnico em Configuração (TC) - é um papel operacional que demanda conhecimentos dos princípios da gerência de configuração. É responsável pela criação, configuração e manutenção das estruturas de dados do projeto e do processo de produção. As alterações de estrutura de objetos na base e carga de dados para testes, também, fazem parte de seu escopo de atuação;
- Testador (T) – é um papel operacional que tem a responsabilidade de executar o software segundo os roteiros e casos de teste especificados e avaliar o resultado obtido. Deve apontar os problemas detectados para que a equipe de desenvolvedores efetue os ajustes necessários.

Uma relação dos atores que compõem a fábrica de software da EMPRESA A com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.6¹⁴.

Figura A.6 – Distribuição dos envolvidos com o processo de produção da EMPRESA A no modelo de categorização por níveis. Nota: A SEPG é um grupo composto por pessoas de ambas visões

Por fim, o processo de produção da fábrica de software da EMPRESA A é caracterizado como pesado, isto é, ele é mais aderente ao *Unified Process* do que ao *eXtreme Program*.

A.1.4 Gestão Operacional

As atividades do processo, apresentadas até o momento, fazem parte da visão processual da organização¹⁵. As atividades pertinentes à visão gerencial

¹⁴ Salienta-se que este trabalho primou por detectar os atores ativos no processo de produção de software, os envolvidos com os demais níveis (estratégico, tático e operacional) foram inseridos no modelo de caracterização proposto por TRINDADE (2006), quando citados pelos entrevistados. Esta informação é válida para todos os casos apresentados neste anexo. É importante salientar que somente as atividades da visão gerencial (a gestão de projeto, da qualidade e de configuração) que se relacionam, diretamente, com a visão operacional serão descritas neste anexo.

(atividades relacionadas à gestão operacional), também foram levantadas, entre elas destacam-se: a Gestão de Projetos, a Gestão de Configuração e; a Gestão da Qualidade.

A atividade de gestão de projetos é responsável pelo planejamento e controle da execução do projeto. Tal atividade é focada no gerenciamento de custo, cronograma, recursos e riscos relacionados ao desenvolvimento de software. O ciclo de execução da atividade de gestão de projetos pode ser definido por meio dos seguintes passos:

1. Um cliente externo, contrata a EMPRESA A para desenvolver um projeto na área de TI.
2. Uma determinada unidade (matriz ou filial) da empresa assume a responsabilidade deste projeto.
3. Quando este projeto possui, em seu escopo, o desenvolvimento de software, a unidade responsável pelo mesmo, pode optar por contratar a fábrica de software da EMPRESA A para desenvolvê-lo (vide Figura A.2).
4. Ao contratar a fábrica, um contato é estabelecido entre o líder de equipe (designado como coordenador de projeto) e o analista de sistemas, tal contato pode ser estendido ao cliente externo, cliente este que contratou a unidade para desenvolvimento do projeto de TI.
5. O coordenador de produção recebe o comunicado oficial sobre o projeto, e um contrato entre o cliente (externo ou interno) e a fábrica de software é estabelecido.

¹⁵ Nota-se que a Figura A.6 foca questões da visão gerencial, inerentes aos envolvidos com o processo de produção. Porém em nenhum momento da descrição do caso foi apresentada as atividades imersas dentro da visão em questão.

6. Ao começar a receber a especificação do software¹⁶, o coordenador de produção inicia o processo de desenvolvimento de infra-estrutura para o projeto, com isto, os seguintes pontos são mapeados:
- a. Tecnologia: Viabiliza a instalação das células tecnológicas para a execução do projeto.
 - b. Estabelecimento da Equipe: O estabelecimento da equipe é feito com base em pontos por função. A contagem dos pontos por função é de responsabilidade do analista de sistemas. Quando esta contagem não está explícita, a mesma é elaborada pelo coordenador de produção, pois ele é quem recebe todo o projeto de arquitetura do analista de sistemas. Quando contados pelo coordenador de produção, os pontos por função necessitam ser validados pelo analista ou pelo cliente externo (quando este é responsável por esta tarefa).
 - c. Treinamento: Quando necessário, o coordenador de projeto solicita ao gerente da fábrica de software um determinado tipo de treinamento para os envolvidos com o projeto.

Por fim, o coordenador de projeto começa a receber a especificação do software¹⁷. A partir daí, o processo de produção da fábrica de software entra em operação, na qual as atividades de equalização, codificação, teste e entrega são percorridas. Um fluxograma retratando todo o procedimento da atividade de gestão de projetos pode ser verificado por da Figura A.7.

Ao analisar tal figura, é possível constatar que um projeto de software é “fatiado” em ordens de serviços (OS). As ordens possuem uma ou mais funcionalidades, já dimensionadas em pontos por função. Com as ordens

¹⁶ Na especificação do software, estão definidas as características do produto, entre elas destacam-se: uma previsão de seu tamanho em pontos por função e o prazo máximo para implantação.

¹⁷ Sabendo que o processo de desenvolvimento de software é incremental, a especificação do projeto é “fatiada”.

pontuadas, o coordenador de produção estabelece uma equipe, formada por um líder e vários desenvolvedores, para executar tal serviço.

Figura A.7 – Diagrama detalhando a atividade de gestão de projeto – EMPRESA A. O círculo preto representa o estado inicial, já o branco representa o estado final.

O gerenciamento da produção é feito sobre a ordem de serviço, sendo assim o coordenador de produção desenvolve uma planilha de acompanhamento do projeto (um artefato), na qual cada funcionalidade é controlada (controle efetuado: cronograma e média de produtividade). Cabe ressaltar que o coordenador de projeto leva em consideração, em seus apontamentos, a capacidade de produção da fábrica de software (média de pontos por função em um determinado período). A data de entrega da ordem de serviço, também, é um fator importante nos apontamentos do coordenador de produção.

Um fato deve ser ressaltado no caso em questão: A EMPRESA A possui uma boa capacidade de dimensionamento, o coordenador de projetos e o gerente de produção da fábrica de software conseguem estimativas como a ilustrada no exemplo a seguir:

"Podemos produzir x pontos por função em um período y ". (Palavras do diretor de produção da fábrica de software da EMPRESA A)

Ao atribuir um serviço a uma equipe, o diretor de produção recebe os apontamentos das funcionalidades que estão sendo desenvolvidas pelos desenvolvedores (percentual de conclusão de uma funcionalidade e horas utilizadas para o desenvolvimento da mesma). Os desenvolvedores para realizar estes apontamentos utilizam a máquina de processo (o detalhamento de tal máquina será apresentado mais adiante). É perceptível no caso, que o desenvolvedor recebe seus vencimentos pelas horas apontadas na máquina.

O principal artefato utilizado pelo coordenador de produção é a planilha de acompanhamento de projeto, esta planilha engloba todas as informações do projeto.

Apresentados os procedimentos sobre gestão de projetos praticados pela EMPRESA A, faz-se necessário, neste momento, apresentar como está configurada a atividade de gestão de configuração.

Tal gestão é realizada de forma simples. Em um primeiro momento é feito um controle de versões de artefatos do processo e de componentes (ou produtos). Em um segundo momento é desenvolvido um controle de produtos disponibilizados aos clientes. A Figura A.8 a apresenta e, nela é possível notar que um componente ou produto possui várias versões, as quais são controladas quanto a mudanças, responsável pela mudança e motivação para mesma. Na figura também é perceptível que os produtos podem ser disponibilizados para os clientes em versões diferentes, informações relacionadas a este fato, também, são controladas pela atividade de gestão de configuração.

Figura A.8 – Esquemático definido para gestão de configuração (EMPRESA A)

Por fim, a gestão da qualidade é desenvolvida a partir das informações geradas pela máquina de processo, principalmente, no que se diz respeito ao controle de erros, mapeados na atividade de teste. Além destas informações, os desvios no processo também servem de parâmetro da qualidade. Reuniões periódicas são realizadas com o objetivo de apresentar o desempenho da equipe de produção da fábrica de software, e propor metas da qualidade a serem atingidas.

A.1.5 Armazenamento de Dados

Base de Ferramentas

Conforme apresentado no capítulo 2, uma fábrica de software deve possuir um conjunto de ferramentas para auxiliar a execução das atividades do processo de produção. A EMPRESA A não foge desta premissa e possui, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das

atividades pertinentes à visão gerencial e processual. A relação das ferramentas com as atividades é mapeada na Tabela A.1¹⁸.

Atividade do processo		Natureza da ferramenta utilizada para automação da atividade	Visão
Equalização		Máquina de processo. Controle de versões. Ferramenta para visualização dos artefatos advindos da atividade de projeto de software. Ferramenta para gestão de projetos.	Processual
Codificação		Ambiente de desenvolvimento de software para JAVA e .NET. Controle de versões. Máquina de processo.	
Teste e Entrega		Máquina de Processo. Geradores de casos de testes. Controle de versões. Ambiente de desenvolvimento de software.	
Gestão Operacional	Projetos	Máquina de processo. Gestão de projetos.	Gerencial
	Configuração	Controle de versões. Máquina de processo.	
	Qualidade	Máquina de processo.	

Tabela A.1 – Relação das ferramentas e atividades do processo – EMPRESA A

É perceptível na Tabela A.1, que a presença da máquina de processo está relacionada a todas as atividades. Este fato se configura devido à definição formal para tal ferramenta (as definições sobre a máquina de processo podem ser verificadas no capítulo 5).

O fato acima explicitado levou o autor deste trabalho a verificar a configuração de tal máquina nas fábricas que se configuram como objetos deste anexo. Na EMPRESA A, a máquina de processo possui as seguintes funcionalidades:

- Manter¹⁹ as atividades do processo de produção;
- Manter as ordens de serviço, derivadas do projeto de software;
- Manter os envolvidos com o processo de produção de software;
- Manter os clientes e seus projetos de software;

¹⁸ A Tabela A.1 apresenta ferramentas caracterizadas como indispensáveis na execução e controle das atividades do processo. Esta informação, também, é válida para os demais casos apresentados neste anexo.

¹⁹ Neste trabalho a palavra manter traduz a idéia de armazenamento, exclusão e consulta de uma determinada informação.

- Manter os produtos desenvolvidos, produtos estes relacionados às ordens de serviços;
- Armazenar os envolvidos, erros e o tempo de desenvolvimento para cada produto em cada versão;
- Relacionar os produtos desenvolvidos aos projetos dos clientes;
- Emitir relatórios gerenciais relacionadas à gestão de projetos e da qualidade.

Além das funcionalidades, uma visão arquitetônica e abrangente²⁰ da máquina de processo é apresentada na Figura A.9.

Figura A.9 – Estrutura da máquina de processo da fábrica de software da EMPRESA A.

Ao analisar a Figura A.9, é possível verificar a presença de 6 módulos. O módulo principal da máquina é o PCP (planejamento e controle de processos). Nele, o gerente de projeto atribui, aos envolvidos com o processo, as atividades de devem ser percorridas para a confecção dos produtos ligados às ordens de serviço.

²⁰ Não foi permitido o acesso à máquina de processo. A visão relatada neste texto foi desenvolvida a partir das considerações efetuadas pelos entrevistados. Esta observação, também, é válida para todas as empresas que compõem este anexo.

Informações gerenciais como a relatada na Tabela A.2, podem ser geradas com a utilização da máquina.

EMPRESA A				
Máquina de processo				
Informações para o Controle da Produção				
Cliente – Empresa XYZ				
Projeto – ABC				
Ordens de serviços: Equalização desenvolvida em 00/00/00	Ordem: 1	Produto – 1.1 Est: Pronto	Envolvidos: José (Líder) João Marcos	Atividade: Cod: Início: 00/00/00 Tér: 00/00/00 h: 3/4 Teste: Início: 00/00/00 Tér: 00/00/00 h: 1/2
		Produto – 1.2	Envolvidos: Maria Pedro	Erros encontrados a partir da atividade de teste: Erro A Erro B
	Ordem: 2			
	Ordem: 3			
Legenda: Est: Estado do Produto = Em desenvolvimento ou Pronto h: n/n1 = h – carga horária: n prevista, n1 realizada.				

Tabela A.2 – Características das Informações geradas pela máquina de processo da EMPRESA A.

Ao analisar a Tabela A.2, é perceptível que os envolvidos com a visão gerencial, recebem uma estimativa de término de confecção dos produtos inerentes a um determinado projeto, a quantidade de horas prevista e a quantidade de horas utilizada para a realização do mesmo. Todo cálculo de estimativa para o planejamento da produção é feito sob a ótica de pontos por função.

Por fim, as informações sobre os projetos encerrados podem ser utilizadas como fonte de planejamento e execução de outro projeto²¹, se este possuir características semelhantes.

Base de Componentes

Conforme relatado na seção inerente a codificação, a fábrica de software da EMPRESA A possui plataformas de desenvolvimento de software definidas para as tecnologias. NET e JAVA. Cabe ressaltar que a fábrica em questão, não possui uma documentação formal, inerente às classes que compõem tal plataforma. Todas as

²¹ Em um ambiente de produção de software, este tipo de planejamento é denominado como planejamento a partir de bases históricas de projetos.

classes definidas na plataforma são classificadas como classes de infra-estrutura (uma classe de infra-estrutura possui o mesmo conceito de componente de infra-estrutura apresentado no capítulo 5), o que possibilita a sua utilização em qualquer tipo de projeto.

A.1.6 Produtos Gerados

Na seção sobre a caracterização da empresa foi possível perceber que soluções relacionadas à Gestão Empresarial (*Enterprise Resource Planning* - ERP); Gerenciamento de Relacionamento com o Cliente (*Customer Relationship Management* - CRM) e Administração Tributária fazem parte da carteira de produtos da EMPRESA A. Em entrevista junto aos membros da empresa, foi questionado se a mesma dá preferência para clientes que desejam somente estes produtos ou se clientes de outros mercados, também, seriam potencializados pela empresa. A resposta foi positiva. Sendo assim, é possível afirmar que a fábrica de software da EMPRESA A não tende a ser orientada a domínios.

Já em relação à tecnologia, foi verificada que a mesma desenvolve produtos utilizando somente JAVA e .NET, sendo assim é possível afirmar que a EMPRESA A é orientada à tecnologia.

Por fim, a orientação a produto traduz a idéia do desenvolvimento e comercialização de um ou dois produtos apenas, por exemplo: uma empresa qualquer comercializa somente o produto X. Fato este que não ocorre com a EMPRESA A.

Seguindo o protocolo, apresentado no capítulo 6, a última informação que deve ser inserida na descrição do caso da EMPRESA A é a forma de criação e organização do processo.

A.1.7 Forma de Criação e Organização do Processo

O processo de produção de software da fábrica de software da empresa em questão foi criado a partir da experiência dos profissionais que nela atuam. “O modelo CMMI serviu como um guia para sua melhoria²²”. A palavra guia é bem empregada, pois aspectos como definição de domínios de aplicações e modelagem tecnológica relacionada à máquina do processo, que são de fundamental importância em um ambiente fabril de produção de software, não são contemplados em tal modelo.

A.2 A Fábrica de Software da EMPRESA B

A.2.1 Caracterização da Empresa

A EMPRESA B está presente no mercado de tecnologia da informação desde 1970, e sua principal missão é prover e melhorar a prestação de serviços que envolva tecnologias de tratamento da informação.

Nacionalmente, a empresa em questão está instalada em nove cidades e, internacionalmente, possui sedes em outros dois países. Com cerca de 4000 colaboradores, a empresa presta serviços de:

- **Sustentação de sistemas:** Segundo a empresa, tipicamente, cerca de 70% dos orçamentos de TI das organizações são destinados a melhorias e atualizações de sistemas, demonstrando, assim, a importância da manutenção de sistemas em todo o ciclo de vida;
- **Transferência e recepção de tecnologia:** Para se manterem atualizadas, em relação à evolução tecnológica e atacar, diretamente, o aumento da competitividade, as organizações procuram direcionar seus recursos para atividades específicas, enquanto transferem outras atividades para companhias

²² Palavras estas proclamadas pelos entrevistados.

parceiras. Com base nesse nicho de mercado, a empresa em questão adotou um conceito de transferência de tecnologia como uma das formas de prestar serviço;

- Integração de sistemas: Sistemas dispersos que necessitam de integração para melhor aplicação da tecnologia aos negócios;
- *Help Desk*: Os serviços são construídos sob os parâmetros do cliente, apoiados por um ponto único de contato para o diagnóstico dos problemas do usuário;
- Fábrica de software: Estrutura de desenvolvimento de software²³ utilizada para automatização dos sistemas de informação.

Cabe ressaltar, ainda, que a EMPRESA B foca o desenvolvimento de soluções que envolvem software para o setor:

- Público – nos âmbitos federal, municipal, estadual e para o setor judiciário e;
- Privado – telecomunicações, energia elétrica, bancos e indústria de cosméticos.

Estruturalmente, a fábrica de software da EMPRESA B está dividida em células de desenvolvimento, cada qual com seu líder. Uma equipe de produção é ligada a cada célula (vide Figura A.10). O SQA é responsável pela garantia da qualidade em relação ao processo de desenvolvimento de software.

Descrita a visão geral da EMPRESA B, o protocolo, descrito no capítulo 6, estabelece que se relate as impressões sobre os insumos do processo de produção, cujo objetivo é caracterizar a fábrica de software dentro dos ciclos de produção: curto, médio ou longo.

²³ A fábrica de software da EMPRESA B foi fundada em 1997. Atualmente a empresa em questão possui certificação CMMI nível 3.

Figura A.10 - Estrutura organizacional da fábrica de software da EMPRESA B.

A.2.2 Insumos do Processo

Em entrevista realizada junto aos membros da empresa, foi possível detectar que o processo de produção de software possui 4 atividades:

- Planejamento: Define, a partir do projeto de software, o tamanho da equipe, o esforço e o tempo para desenvolvimento;
- Construção: Dividida em 3 tarefas:
 - Análise ou Equalização;
 - Codificação;
 - Controle da qualidade: Na fábrica de software da EMPRESA B, esta tarefa está, intimamente, ligada ao conceito de teste.
- Encerramento: Armazena as lições aprendidas com o projeto;

- Garantia: Atividade ligada ao conceito de manutenção do produto após sua entrega.

Além das atividades pertinentes ao processo fabril, a EMPRESA B possui uma a fábrica de projetos, cuja responsabilidade é executar as atividades de modelagem de negócio e projeto de software, ficando a cargo da fábrica de software somente as atividades inerentes ao ciclo curto (equalização, codificação e teste). Uma relação entre as fábricas, de software e de projetos, é mapeada na Figura A.11. Ao analisar tal figura, é perceptível que a fábrica de projetos é denominada, pela EMPRESA B, como citoplasma, já a fábrica de software se configura como o núcleo da estrutura de produção. Na parte inferior da figura, é perceptível que a fábrica de projetos contrata a fábrica de software para a execução das atividades codificação e teste, sendo que a porta de equalização é responsável por verificar a “completude” e a “corretude” dos artefatos que irão adentrar o ciclo de produção da fábrica de software.

Figura A.11 – Modelo de produção da EMPRESA B

Baseado na configuração apresentada pela Figura A.11, é possível concluir que EMPRESA B caracteriza sua produção de software dentro do ciclo longo, já a fábrica de software, unidade visitada para a realização do estudo de caso, direciona sua produção dentro do ciclo curto.

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção.

A.2.3 Atividades do Processo

Conforme relatado, anteriormente, as atividades pertinentes ao processo de software da EMPRESA B são: modelagem de negócio, projeto de software, planejamento, construção (dividida em equalização, codificação e teste), encerramento e garantia. A fábrica de software executa somente as atividades de planejamento, construção, encerramento e manutenções corretivas. É importante ressaltar que as melhorias do produto são encaradas como um novo projeto, e estas devem ser especificadas pela fábrica de projetos.

A Atividade de Construção/Equalização

A atividade de equalização (vide Figura A.12) tem como objetivo verificar a “corretude” e a “completude” das especificações de projeto de software para que as mesmas possam ser colocadas em linha de produção (codificação e teste). A execução desta atividade envolve a presença do analista de sistemas e do líder de célula.

Na execução da atividade em questão, o líder de célula se reúne com o analista de sistemas para estabelecer um entendimento comum sobre os artefatos do projeto. Salienta-se que tais artefatos devem respeitar o padrão pré-estabelecido pela empresa. As principais saídas da atividade equalização são:

- a divisão do projeto em pacotes e ordens de serviços (OS), respeitando sempre a estrutura tecnológica;

Figura A.12 – Atividades da fábrica de software da EMPRESA B. Nota: A implantação faz parte da atividade de encerramento e a manutenção faz parte da atividade de garantia

- configuração dos tamanhos²⁴ das ordens de serviços, e;
- as configurações inerentes aos casos e dados de testes que serão utilizados em tal atividade.

Alguns artefatos pertinentes à atividade de equalização foram inferidos a partir das entrevistas realizadas, entre eles destacam-se:

- Roteiro para realização da equalização: Documento que auxilia tanto o analista de sistemas, quanto o líder de célula em tal atividade;

²⁴ A unidade de medida utilizada pela fábrica de software da EMPRESA B é pontos por função. A medida de esforço é configurada em horas de trabalho.

- Padronização para definir o relatório de teste: Os relatórios de testes serão utilizados na atividade de controle da qualidade, estes devem ser padronizados em relação ao formato e ao conteúdo. Cabe ressaltar que o relatório é um artefato aglutinador e possui em sua estrutura casos e dados de testes para serem testados;
- Formulário de gestão de projetos²⁵: Artefato que aglomera os pacotes, as ordens de serviço e, o tamanho, em pontos por função, de cada OS.

A Atividade de Construção/Codificação

A atividade de codificação da fábrica de software da EMPRESA B é desenvolvida sobre a ordem de serviço, respeitando a tecnologia na qual o projeto está imerso. Por exemplo: projetos que possuem em seu escopo a tecnologia JAVA são desenvolvidos por profissionais que possuem familiaridade com ela. Salienta-se que a fábrica de software não possui uma biblioteca de componentes ou uma plataforma de desenvolvimento formalizada e o reuso de código é praticado, informalmente.

Dentro da atividade de codificação, o desenvolvedor utiliza a máquina de processo para informar a sua produtividade, fato também presente no estudo de caso realizado na fábrica de software da EMPRESA A.

Por fim, alguns artefatos utilizados nesta atividade são destacados:

- padrão para desenvolvimento de código definido para cada linguagem;
- e o formulário de apontamento de produtividade, pertinente à máquina de processo.

²⁵ O formulário de gestão de projetos também é utilizado na atividade de planejamento.

A Atividade de Construção/Controle da Qualidade ou Teste

A atividade de teste²⁶ de software tem como objetivo verificar a qualidade dos produtos gerados pela fábrica. Salienta-se que tal atividade está dividida em três tarefas:

- Teste unitário do componente ou da funcionalidade.
- Teste de Integração dos componentes ou das funcionalidades.
- Teste de aceitação desenvolvido na presença do cliente. Teste realizado pela fábrica de projetos.

Na fábrica de software da EMPRESA A os testes unitários e de integração são realizados por dois desenvolvedores²⁷, o que codificou e um convidado que não participou da codificação da ordem de serviço. Caso um componente ou um grupo de componentes apresente problemas, os mesmos são codificados novamente, em uma ordem de serviço corretiva, gerando uma nova entrada na máquina de processo²⁸, com isto, novos testes: unitário e de integração, são desenvolvidos.

Por fim, é necessário destacar a presença da atividade de manutenção corretiva dentro do processo de produção da fábrica de software da EMPRESA B. Tal atividade é executada, quando algum erro (o produto não atendeu, plenamente, o requisito estabelecido no projeto ou possui problemas em relação à codificação) ocorre com o produto instalado no cliente, os dados, como tempo e esforço, empregados nas correções, são registrados na máquina de processo para, posteriormente, servirem como estatísticas para análise da qualidade.

²⁶ A atividade de teste utiliza o relatório de teste definido na atividade de equalização.

²⁷ A atividade de teste da fábrica de software da EMPRESA B foi concebida sob a ótica da revisão em pares.

²⁸ Os dados sobre as ordens de serviços corretivas, armazenados na máquina de processo, são utilizados para fins estatísticos no processo de controle da qualidade da fábrica de software da empresa em questão.

Após a apresentação das atividades operacionais relacionadas ao processo de produção da fábrica de software, o roteiro estabelecido na Tabela 6.3 prevê que seja descrito, no relato do caso, o modelo de processo utilizado.

A EMPRESA B utiliza modelo incremental. Este fato é perceptível ao analisar a figura do Analista de Sistema, cuja responsabilidade é colher os requisitos do produto junto ao cliente e desenvolver o modelo de negócio. Tal modelo é projetado, na forma de um software. O projeto de software é “fatiado” em ordens de serviços (Figura A.12). As ordens de serviços são codificadas, testadas, gradualmente, caracterizando assim o aspecto de “incrementabilidade” do processo.

Uma outra constatação detectada é a participação dos atores (envolvidos) na execução das atividades do processo. Na descrição das atividades processuais e gerencias, é possível verificar a presença do:

- Analista de Sistemas (AS): Descrição semelhante à apresentada na seção que relata o caso da EMPRESA A.
- Desenvolvedor (D) – Possui todas as responsabilidades descritas na seção que retrata o caso da EMPRESA A e trabalha, ativamente, no processo de equalização dos artefatos;
- Garantia da Qualidade de Software (Software Quality Assurance - SQA) – Descrição idêntica à apresentada na seção que relata o caso da EMPRESAS A;
- Gerente da Fábrica de Software (GFS) – é um papel gerencial dentro da empresa, com a responsabilidade de avaliar novos projetos, elaborar propostas técnicas e acompanhar a execução de todos os projetos em andamento, de forma que cada software esteja de acordo com as especificações mapeadas, e que sejam produzidos com qualidade, nos prazos e dentro do orçamento estabelecido;

- Gerente de Configuração (GC) - é um papel gerencial que demanda conhecimentos nos princípios da gerência de configuração. É responsável pela criação, configuração e manutenção das estruturas de dados do processo de produção. As alterações de estrutura de objetos, na base de dados, e a carga de dados para testes, também, fazem parte de seu escopo de atuação;
- Líder de Célula (LC) – Descrição semelhante à do líder de equipe da EMPRESA A (vide seção A.1);
- SEPG - é um grupo formado pelos gerentes da fábrica, SQA e alguns líderes de células que têm como objetivo definir, acompanhar e avaliar o processo de software.

Uma relação dos atores que compõem a EMPRESA B com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.13

Figura A.13 – Distribuição dos envolvidos com o processo de produção da EMPRESA B no modelo de categorização por níveis. Nota: A SEPG é um grupo composto por pessoas de ambas visões. O Gerente da fábrica de Software atua nos níveis tático e operacional.

Por fim, o processo de produção da fábrica de software da EMPRESA B é caracterizado como pesado, isto é, ele é mais aderente ao *Unified Process* do que ao *eXtreme Program*.

A.2.4 Gestão Operacional

As atividades do processo, apresentadas até ao momento, fazem parte da visão processual da organização. As atividades de gerenciamento de projetos da qualidade e de configuração, que pertencem à visão gerencial, também foram mapeadas no estudo de caso em questão.

A atividade de gerenciamento é responsável por desenvolver o planejamento do projeto. Questões como tamanho do software a ser desenvolvido, baseado nas especificações aludidas pela fábrica de projetos, a medida de esforço, o tamanho da equipe, o cronograma, a infra-estrutura tecnológica, risco e a matriz de comunicação²⁹, são definidos em tal atividade.

Para efetuar o planejamento do projeto, o gerente da fábrica de software conta com o auxílio dos líderes de células. Tais líderes, de posse das especificações desenvolvidas pela fábrica de projetos, consultam a base histórica³⁰, estabelecem comparações entre as especificações e o projetos terminados, desenvolvendo, assim, a atividade planejamento. O gerente da fábrica acompanha o trabalho dos líderes e pode aprovar ou reprovar o plano de projeto desenvolvido.

Cabe ressaltar que na atividade de planejamento a métrica utilizada é pontos por função. Ela é aplicada sobre as ordens de serviços, visto que o projeto de software é “fatiado” na atividade de equalização. A Figura A.14 apresenta a atividade de planejamento realizada pela fábrica de software da EMPRESA B.

²⁹ A matriz de comunicação estabelece permissões de comunicação entre os envolvidos com o processo de produção de software, por exemplo: em um projeto qualquer fica definido que a única pessoa que pode se comunicar com o cliente é o analista de sistemas.

³⁰ A base histórica de projetos é alimentada pela máquina de processo.

Figura A.14 – Atividade de planejamento da fábrica de software (EMPRESA B). PLN: Planejamento. As demais siglas estão definidas na Figura A.12

Ao analisar a figura acima é possível perceber que o planejamento é executado após a atividade de equalização e tem, como principal artefato de saída o plano de projeto. O controle ou acompanhamento do projeto tem como objetivo verificar se existe algum desvio no plano, se estes forem encontrados, correções são aplicadas, delineando, assim, o conceito de replanejamento. Por fim, na figura estão presentes os envolvidos na atividade em questão: líder de célula e o gerente da fábrica.

A atividade de encerramento de projeto também faz parte da visão gerencial da fábrica de software da EMPRESA B. Nesta atividade é desenvolvida uma ata sobre o projeto, nela são registrados os dados estatísticos inerentes a execução do projeto e as lições aprendidas³¹.

Por fim, as atividades de gerenciamento de configuração e da qualidade são semelhantes às realizadas pela fábrica de software da EMPRESA A.

³¹ Tais dados são armazenados na base de histórica e utilizados em projetos futuros.

A.2.5 Armazenamento de Dados

Base de Ferramentas

A fábrica de software da EMPRESA B possui em sua estrutura um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão processual e gerencial. A relação das ferramentas com as atividade são apresentadas na Tabela A.3.

Atividade do processo		Natureza da ferramenta utilizada para automação da atividade	Visão
Equalização		Ferramentas que foram utilizadas para gerar os artefatos inerentes à atividade de projeto de software.	Processual
Codificação		Ambiente de desenvolvimento de software nas mais variadas plataformas. Controle de versões. Máquina de processo.	
Controle da Qualidade (Teste)		Máquina de Processo. Geradores de casos de testes. Controle de versões. Ambiente de desenvolvimento de software	
Garantia (Manutenção corretiva)		Ambiente de desenvolvimento de software. Controle de versões. Máquina de processo.	
Gestão Operacional	Projetos	Máquina de Processo. Gestão de projetos.	Gerencial
	Configuração	Controle de Versões. Máquina de Processo	
	Qualidade	Máquina de Processo	
Encerramento		Máquina de Processo	

Tabela A.3 – Relação das ferramentas e atividades do processo – EMPRESA B

As mesmas considerações, sobre a máquina de processo, efetuadas para a fábrica de software da EMPRESA A, também são válidas para a fábrica de software da EMPRESA B.

Base de Componentes

Conforme relatado, anteriormente, o reuso de código ou componente, efetuado pela fábrica de código da EMPRESA B, é realizado informalmente.

A.2.6 Produtos Gerados

Na seção sobre a caracterização da empresa foi possível perceber que soluções relacionadas ao setor público (federal, municipal, estadual e para o setor judiciário) e ao setor privado (telecomunicações, energia elétrica, bancos e industria

de cosméticos) fazem parte da carteira de produtos da EMPRESA B. Em entrevista junto aos membros da empresa, foi questionada se a mesma dá preferência para clientes que desejam somente este tipo de produto ou se clientes de outros mercados também seriam potencializados pela empresa. A resposta foi positiva. Sendo assim, não é possível afirmar que a fábrica de software da EMPRESA B seja orientada a domínios.

Já em relação à tecnologia, foi verificado que a fábrica desenvolve produtos utilizando em várias tecnologias (de COBOL a JAVA). Sendo assim, é possível afirmar que a fábrica de software da EMPRESA B não é orientada à tecnologia.

Por fim, a orientação a produto traduz a idéia do desenvolvimento e comercialização de um ou dois produtos apenas, por exemplo: uma empresa qualquer comercializa somente o produto X. Fato este que não ocorre com a EMPRESA B.

A.2.7 Forma de Criação e Organização do Processo

As considerações, sobre a forma de criação e organização do processo de produção de software, efetuados no estudo de caso da EMPRESA B são semelhantes às da EMPRESA A.

A.3 A Fábrica de Software da EMPRESA C

A.3.1 Caracterização da Empresa

A EMPRESA C atua na área de Tecnologia da Informação (TI) desde 1989 e possui uma ampla carteira de produtos e serviços. Atualmente, a empresa se destaca na prestação de serviços e desenvolvimento de produtos nas seguintes áreas:

- Engenharia de processos: Atividade direcionada à modelagem do negócio em nível empresarial, focando a otimização de três elementos: pessoas, tecnologia e informação;
- Desenvolvimento de software: Atividade que foca a automatização de sistemas de informação para as empresas nos mais variados domínios do conhecimento. Esta atividade é apoiada pelas fábricas de projetos, de código e de testes;
- Manutenção de sistemas: Atividade que foca em correções emergenciais nos sistemas automatizados;
- Gestão de ambiente e infra-estrutura: Atividade com o objetivo de absorver a terceirização da área de TI de uma organização, incluindo o suporte técnico de hardware, banco de dados, servidores e *help desk*;
- Produtos: Desenvolvimento de soluções para treinamento à distância, gestão bancária, ERP, *Business Intelligence* (BI), gestão hospitalar e gestão eletrônica de documentos.

Atualmente, a empresa conta com cerca de 8000 colaboradores, espalhados nos mais variados projetos por todo o território nacional. A empresa em questão conta com oito unidades instaladas, 3 delas localizadas no estado de São Paulo. Paraná, Rio de Janeiro, Distrito Federal, Minas Gerais e Sergipe completam o restante das unidades.

A principal missão da EMPRESA C é fornecer, a seus clientes, vantagens competitivas, prestando serviços de valor agregado, com elevados padrões de qualidade e produtividade.

Salienta-se que a EMPRESA C não divulgou a sua estrutura organizacional na coleta dos dados.

Descrita a visão geral da EMPRESA C, o protocolo, descrito no capítulo 6, estabelece que se relate as impressões sobre os insumos do processo de produção, cujo objetivo é caracterizar a fábrica de software dentro dos ciclos de produção: curto, médio ou longo.

A.3.2 Insumos do Processo

A fábrica de software da EMPRESA C está dividida em três unidades, denominadas fábrica de projeto, fábrica de código e fábrica de teste.

A fábrica de projetos tem como objetivo desenvolver a modelagem de negócio (ou análise de sistemas), o projeto e a implantação do software.

Já a fábrica de código tem como meta principal, realizar a codificação dos artefatos gerados na atividade de projeto de software. No processo de codificação, os artefatos gerados pela fábrica de projetos, são equalizados (artefatos devem ser entendidos pelos envolvidos com a fábrica de código). Além das atividades de equalização e codificação, o teste é realizado, informalmente, dentro da fábrica de código.

Por fim, a fábrica de teste, de posse das unidades de software geradas³², realiza, formalmente, o teste unitário e de integração.

A Figura A.15 representa, arquitetonicamente, a configuração do modelo de produção da fábrica de software da EMPRESA C. Por meio da figura é perceptível que o produto software está envolvido pelas 3 fábricas. Salienta-se que a linha de gestão de projetos permeia todas as fábricas, fato este que nos leva a crer que um projeto é gerenciado, independentemente, de seu momento de produção, o gerenciamento é realizado, paralelamente, na modelagem de negócio e projeto de

³² Para a EMPRESA C uma unidade de software pode ser caracterizada como uma funcionalidade de um determinado produto de software, ou como um componente de infra-estrutura que será compartilhado por vários produtos.

software, na codificação e nos testes. Lembrando sempre que questões contratuais firmadas com os clientes devem ser atendidas pelas fábricas.

Em relação ao ciclo de produção, a EMPRESA C está contextualizada dentro do ciclo longo, desde que as três fábricas estejam operando em conjunto para um determinado projeto. Para se contextualizar sob a ótica do ciclo médio, a fábrica de código deve operar em conjunto com a fábrica de teste e executar a atividade de projeto de software. Já a contextualização do ciclo curto dentro do processo de produção da EMPRESA C, ocorre quando a fábrica de código opera de forma paralela a fábrica de teste.

Figura A.15 – Modelo de produção da EMPRESA C

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção. Salienta-se que as informações, apresentadas nela, estão alicerçadas sobre as fábricas de código e de testes.

A.3.3 Atividades do Processo

As atividades do processo de produção de software das fábricas em questão são: Equalização, Codificação, Teste Unitário e Teste Integrado.

A Atividade de Equalização

A atividade de equalização tem como objetivo prover o entendimento dos artefatos advindos da atividade de projeto de software, para que os mesmos possam ser colocados em linha de produção (codificação e teste).

A dinâmica da atividade em questão ocorre em torno de várias reuniões³³ de trabalho, nas quais participam o analista de sistemas (envolvido com a fábrica de projeto) e o líder de projeto (envolvido com a fábrica de código). Nestas reuniões são sanadas as dúvidas que recorrem sobre os artefatos gerados na atividade de projeto de software. Após a realização destas reuniões, a ata de equalização (a ata de equalização é considerada um artefato - ou ativo do processo) é preenchida e assinada pelos participantes. Nesta ata são registrados os problemas (dúvidas) e as soluções inferidas durante a execução da atividade de equalização.

A Atividade de Codificação

A codificação da fábrica é desenvolvida sobre o conceito de ordens de serviço, amplamente discutido nos casos anteriores, respeitando a tecnologia e o banco de dados na qual o projeto está imerso. Por exemplo: projetos que possuem em seu escopo a linguagem JAVA e necessitam acessar banco de dados Oracle (KOLSTE e PETERSEN (1995)) são desenvolvidos por profissionais que possuem familiaridade com tais produtos. Salienta-se que a fábrica de software não possui

³³ O termo várias reuniões, foi empregado pelos entrevistados, devido ao modelo de processo utilizado. Tal modelo é caracterizado como incremental, sendo assim, faz-se necessário a realização de uma reunião toda vez que ocorrer incrementos no projeto.

uma biblioteca de componentes ou uma plataforma de desenvolvimento formalizada, o reuso de código é aplicado, informalmente.

Dentro da atividade de codificação, o desenvolver utiliza a máquina de processo para informar a sua produtividade, fato que também ocorre junto as demais empresas.

Por fim, os artefatos utilizados na atividade de codificação são: padrão para desenvolvimento de código e o formulário de apontamento de produtividade, pertinente à máquina de processo.

A Atividade de teste

A atividade de teste é regida pela fábrica de teste. Esta estrutura tem como objetivo desenvolver casos e dados de testes de acordo com os projetos em execução dentro da EMPRESA C. No desenvolvimento dos casos e dos dados, os envolvidos com a fábrica de teste entram em contato com a fábrica de projetos para capturar os requisitos funcionais mapeados dentro do escopo do produto. Conforme apresentado na Figura A.15, a fábrica de teste realiza o:

- Teste unitário: o componente ou a funcionalidade é testado, utilizando os dados e casos de testes, previamente, definidos;
- Teste de Integração: Os componentes ou as funcionalidades são integrados e um teste é desenvolvido, utilizando os dados e casos de testes, previamente, definidos.

O teste de aceitação, tarefa imersa na atividade de implantação pertinente a fábrica de projetos, é desenvolvido na presença do cliente, sendo assim, a fábrica de teste está isenta da realização de tal trabalho.

No teste unitário e de integração, quando erros são detectados, uma nova ordem de serviço, denominada de ordem de serviço de correção de erros, é gerada pela fábrica de teste. Nela estão contidos os dados, os casos de testes utilizados e os erros encontrados nas funcionalidades e nos componentes testados. A ordem de serviço em questão é enviada para a fábrica de código, para que os problemas encontrados nos ativos de software (ou funcionalidades) sejam resolvidos.

Todas informações geradas com a execução das atividades, dentro da fábrica de teste, são coletadas pela máquina de processo e armazenadas em uma base de dados. Estas informações são utilizadas, posteriormente, para o gerenciamento e controle da qualidade da fábrica de software (neste caso, as fábricas, de código e de teste).

De posse da descrição das atividades do processo, e com base no protocolo, descrito no capítulo 6, faz-se necessário neste momento, apresentar as informações inerentes ao modelo de processo.

Por meio das entrevistas realizadas foi constatado que a EMPRESA C utiliza, como as demais empresas, o modelo de processo incremental, com isto, é válido afirmar que as considerações efetuadas sobre este fato para tais empresas também são válidas para a EMPRESA C.

Uma outra constatação detectada é a participação dos atores (envolvidos) na execução das atividades do processo. Na descrição das atividades processuais e gerenciais, é possível verificar a presença do:

- Analista de Sistemas (AS): Descrição semelhante a apresentada na seção que relata o caso da EMPRESA A.
- Desenvolvedor (D) - é um papel operacional e direcionado para o domínio de técnicas de programação e ferramentas de desenvolvimento, com a

responsabilidade de equalizar e implementar as unidades de software (que forem especificadas para produção);

- Garantia da Qualidade de Software (Software Quality Assurance - SQA) – Descrição idêntica à apresentada nas seções que relatam os casos das EMPRESAS A e B;
- Gerente de Produção (GP) – é um papel gerencial dentro da empresa, com a responsabilidade de gerenciar toda a produção da fábrica de código e de teste, acompanhar a execução de todos os projetos em andamento, de forma que cada software esteja de acordo com a especificação, advinda da atividade de projeto. Aspectos relacionados à qualidade do produto, prazos e orçamento são controlados por tal gerente;
- Líder de Projeto (LP) – Descrição semelhante a do Líder de Equipe da EMPRESA A (vide seção A.1);
- SEPG - Descrição semelhante à apresentada nas seções que relatam os casos das EMPRESAS A e B;
- Técnico em Configuração (TC) - Descrição semelhante à apresentada nas seções que relata o caso da EMPRESA A;
- Testador (T) – é um papel operacional que tem a responsabilidade de projetar os casos de testes e os dados de teste (unitário e de integração), executar os testes e informar à fábrica de código sobre a ocorrência de possíveis erros.

Uma relação dos atores que compõem a EMPRESA C com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.16.

Figura A.16 – Distribuição dos envolvidos com o processo de produção da EMPRESA C no modelo de categorização por níveis.

Por fim, o processo de produção da fábrica de software da EMPRESA C é caracterizado como pesado, isto é, ele é mais aderente ao *Unified Process* do que ao *eXtreme Program*.

A.3.4 Gestão Operacional

As atividades do processo, apresentadas até o momento, fazem parte da visão processual da organização. As atividades de gerenciamento de projetos, da qualidade e de configuração, que pertencem à visão gerencial, também foram mapeadas no estudo de caso em questão.

A atividade de gerenciamento de projetos da EMPRESA C é dividida em 3 tarefas:

- **Planejamento:** Estruturalmente, o planejamento é embasado pelo conceito de bases históricas de projetos e pontos por função. Tal atividade é desenvolvida após o levantamento de parte dos requisitos do software. Cabe ressaltar que o levantamento de requisitos, desenvolvido pela fábrica de projetos, ocorre, concorrentemente, com a modelagem de negócio e com o projeto de software. A fábrica de código, unidade visitada para o desenvolvimento deste estudo, recebe o projeto de software “fatiado” em ordens de serviços. De posse de tais ordens o cronograma de execução do projeto é definido, os recursos necessários (equipamentos, aplicativos³⁴ e humanos) são mapeados, o custo para execução do projeto é calculado e, por fim, os riscos são analisados. Salienta-se que existe uma negociação entre a fábrica de projetos e a fábrica de código, pois é a fábrica de projetos que assume compromissos com o cliente, com isto, é necessário que a fábrica de código, encarada como uma subcontratada da fábrica de projetos, também esteja disposta a cumprir as metas e prazos estipulados pela fábrica de projetos;
- **Execução:** Nesta tarefa as atividades do processo de produção são executadas;
- **Controle:** Tarefa responsável por verificar se o que foi planejamento está sendo cumprido. Cabe ressaltar que se o plano de projeto possuir algum desvio, ajustes devem ser efetuados;

Uma visão ilustrativa da atividade de planejamento de projeto, da fábrica de software em questão, pode ser verificada por meio da Figura A.17.

³⁴ Exemplos de aplicativos: compiladores, ambientes de desenvolvimento de software, ferramentas computacionais para gestão de projetos.

Figura A.17 – Atividade de gestão de projetos da fábrica de código e de teste da EMPRESA C.

Por fim, as atividades de gestão de configuração e gestão da qualidade são semelhantes às realizadas pela fábrica de software da EMPRESA A.

A.3.5 Armazenamento de Dados

Base de Ferramentas

A EMPRESA C possui em sua estrutura um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão gerencial e processual. A relação das ferramentas com as atividades são apresentadas na Tabela A.4.

Atividade do processo		Natureza da ferramenta utilizada para automação da atividade	Visão
Fábrica de Código	Equalização	Ferramentas utilizadas no desenvolvimento dos artefatos do projeto de software	Processual
	Codificação	Ambiente de desenvolvimento e banco de dados (de acordo com a necessidade do projeto a ser desenvolvido). Máquina de Processo. Controle de Versões	
Fábrica de teste	Teste Unitário	Máquina de Processo. Geradores de casos de testes. Controle de versões.	
	Teste de Integrado	Máquina de Processo. Geradores de casos de testes. Controle de versões. Ambiente de desenvolvimento de software.	
Gestão Operacional	Projetos	Máquina de processo. Gestão de projetos.	Gerencial
	Configuração	Controle de Versões. Máquina de Processo	
	Qualidade	Máquina de Processo	

Tabela A.4 - Relação das ferramentas e atividades do processo – EMPRESA C

Por fim, as considerações, sobre a máquina de processo, efetuadas para às fábricas de software das EMPRESAS A e B são válidas para as fábricas, de código e de testes, da EMPRESA C.

Base de Componentes

O reuso de código ou componente, efetuado pela fábrica de código da EMPRESA C, é realizado, informalmente.

A.3.6 Produtos Gerados

Na seção sobre a caracterização da empresa foi possível perceber que a EMPRESA C possui uma área de desenvolvimento de software que atende os mais variados domínios do conhecimento. Sendo assim, é possível afirmar que a EMPRESA C não tende a ser orientada a domínios.

Já em relação a tecnologia, foi verificada que a mesma desenvolve produtos utilizando as tecnologias, JAVA, .NET e Oracle, sendo assim é possível afirmar que a fábrica de software da EMPRESA C tende a ser orientada a tecnologia.

Por fim, a orientação a produto, traduz a idéia do desenvolvimento e comercialização de um ou dois produtos apenas, por exemplo: uma empresa qualquer comercializa somente o produto X. Fato este que não ocorre com a EMPRESA C.

A.3.7 Forma de Criação e Organização do Processo

As considerações mapeadas nos casos das EMPRESAS A e B são semelhantes as da EMPRESA C.

A.4 A Fábrica de Software da EMPRESA D

A.4.1 Caracterização da Empresa

A EMPRESA D presta serviço na área Tecnologia da Informação (TI) a cerca de 42 anos, e seu principal objetivo é prover soluções que potencializem o negócio de seus clientes, utilizando um aparato tecnológico competitivo e inovador. Atualmente, a empresa conta com 35.000 clientes e 120.000 funcionários (cerca de 33.000 engenheiros de softwares) espalhados pelos cinco continentes.

A empresa em questão está dividida em 4 grandes áreas:

- Consultoria de negócio: Desenvolvimento e alinhamento do processo de negócio de seus clientes com o aparato tecnológico existente.
- Infra-estrutura: Manutenção e suporte técnico a servidores de redes e voz sobre IP (*Internet Protocol*).
- Aplicações: Desenvolvimento de softwares e manutenção/modernização de sistemas legados, migração da plataforma *mainframe* para a plataforma PC (*personal computer*).
- Gerenciamento de processos de negócio que não sejam o *core business* das empresas, por exemplo: *call center* de uma empresa de cartão de crédito (o *core business* de uma empresa que trabalha com esta atividade é o CRM).

Com uma estrutura organizacional focada no cliente, a empresa em questão, possui duas visões organizacionais, denominadas, por ela própria, como *front-end* e *back-office*.

A visão *front-end* atua, diretamente, com os clientes, caracterizados como o eixo motriz da empresa. Para atender a um determinado cliente, a empresa possui um forte conceito de gestão de relacionamento, que incorpora a venda de produtos e serviços. Paralelamente, a tal gestão, a empresa prevê que a área de engenharia de domínio, cujo objetivo é maximizar os produtos e serviços adquiridos por um determinado cliente, no âmbito de sua organização, supra todas as necessidades dos clientes potencializados pela área de gestão de relacionamentos. Por fim, o nível organizacional em questão, conta com uma área de contratação, na qual esta provê o acesso ao nível organizacional *back-office*. Este nível opera como provedor de serviços e produtos (consultoria de negócio, infra-estrutura, aplicações e gerenciamento de processos de negócio).

A Figura A.18 apresenta, de forma ampla, a estrutura organizacional da EMPRESA D. Em tal figura é perceptível também, dentro da visão de *back-office*, as unidades integradoras, responsáveis por distribuir parte da prestação de serviços e do desenvolvimento de produtos para as unidades de produção. Estas unidades são responsáveis pelo desenvolvimento dos produtos e serviços, pertinentes às atividades de atuação da EMPRESA D.

Figura A.18 – Estrutura organizacional da EMPRESA D

Ressalta-se que este caso foi desenvolvido sob a luz da área de aplicações, mais precisamente, no setor de desenvolvimento de software. Tal área congrega as unidades integradoras e de produção localizadas nos Estados Unidos, na América Latina, no Canadá, na África e Oriente Médio, na região do Oceano Pacífico e da Ásia e na Austrália.

Ainda dentro do contexto organizacional, é possível ressaltar que a EMPRESA D estabelece que todas as unidades (integradora e de produção) utilizem o mesmo modelo funcional, compartilhando as metodologias de desenvolvimento de software e a infra-estrutura utilizada para a gestão de projetos. Dentro deste modelo é possível encontrar a área “gerenciamento estratégico de capacidades” do processo de produção de software.

É importante ressaltar, ainda, que as informações inseridas, nesta seção, derivam de uma visita realizada a uma unidade de produção, localizada na região sudeste do Brasil. A unidade visitada é caracterizada como fábrica de software e em relação aos aspectos da qualidade possui certificação CMMI nível 5.

Descrita a visão geral da EMPRESA D, o protocolo, descrito no capítulo 6, estabelece que se relate as impressões sobre os insumos do processo de produção, cujo objetivo é caracterizar a fábrica de software dentro dos ciclos de produção: curto, médio ou longo.

A.4.2 Insumos do Processo

No estudo de caso foi possível constatar que a área de “gerenciamento estratégico de capacidades” (vide seção A.4.1) definiu que a fábrica de software, visitada, opera sob a ótica do ciclo curto, para determinados projetos e sob a ótica do ciclo médio para outros projetos.

Dentro das atividades relacionadas ao ciclo curto é possível destacar: a equalização do projeto de software, a produção do código e o teste software³⁵. Nesta configuração, o modelo de negócio (quando a fábrica de software opera nos ciclos, médio e curto) e o projeto de software (quando a fábrica opera somente no ciclo curto) são definidos pelo cliente externo, caracterizado como uma subsidiária da EMPRESA D, que ficou com a responsabilidade de realizar tal atividade. Com isto, constata-se que a fábrica de software operante no ciclo curto passa a ser uma subcontrata de uma unidade integradora ou de uma unidade de produção³⁶, a qual possui a responsabilidade de operar nos ciclos, longo ou médio. Uma visão da configuração dos ciclos de produção, presente na fábrica de software, em questão, pode ser verificada por meio da Figura A.19 e Figura A.20.

Figura A.19 – Configuração do ciclo curto de produção da unidade de produção de software, caracterizada como fábrica (EMPRESA D)

³⁵ Atividades de cunho gerencial também são desenvolvidas pela unidade de produção visitada, entre tais atividades é possível destacar o gerenciamento de projetos, gerenciamento de configuração e garantia da qualidade do produto e do processo. Tais atividades serão descritas em uma seção específica.

³⁶ A unidade de produção visitada presta serviço para as unidades localizadas na América do Norte.

Figura A.20 - Configuração do ciclo médio de produção da unidade de produção de software, caracterizada como fábrica (EMPRESA D). Destaca-se a presença da atividade equalizar modelo de negócio, cujo objetivo é configurar as características de “corretude” e “completude” ao engenheiro de software, que atua como projetista.

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção.

A.4.3 Atividades do Processo

Analisando a configuração dos ciclos de produção da fábrica de software, imersa no ambiente organizacional da EMPRESA D, é possível constatar a presença das atividades: equalização do modelo de negócio, projetar software, equalizar projeto de software, produzir código e testar software.

Excluindo a atividade de teste, os entrevistados não divulgaram informações aprofundadas sobre os procedimentos e métodos utilizados na execução das atividades que compõem o processo de produção. Este fato levou o relator do caso a apresentar, apenas, uma visão superficial de cada uma das atividades, mediante as considerações efetuadas pelos entrevistados.

Equalização do Modelo de Negócio

A equalização do modelo de negócios tem como objetivo verificar as especificações do modelo de negócio quanto à “completude”, “corretude” e consistência em relação aos requisitos do cliente, bem como assegurar seu entendimento, para que as mesmas possam ser colocadas em linha de produção (atingindo a atividade de projeto de software), mitigando os riscos de parada na produção e quebra na produtividade. A equalização da modelagem de negócio é uma atividade optativa e deve ser realizada quando as atividades de modelagem do negócio e projeto de software são realizadas por pessoas diferentes.

Projeto de Software

A atividade de projeto de software tem como objetivo desenvolver a arquitetura do produto em questão, a estrutura funcional e os requisitos não funcionais do software são mapeados nesta atividade. Na atividade de projeto, da fábrica de software em questão, alguns artefatos foram detectados, entre eles destacam-se: diagramas de casos de uso, de classes, seqüências e de atividades.

Equalização do projeto de software

A equalização dos artefatos do projeto de software tem como objetivo verificar se os artefatos advindos da atividade de projeto são padronizados e compreensíveis.

Codificação (Produzir software)

Atividade que tem como objetivo desenvolver ou codificar o projeto de software dentro de um conjunto de padrões tecnológicos estabelecidos pela unidade contratante. Os envolvidos com esta atividade utilizam o padrão para o desenvolvimento de código desenvolvido pela área de “gerenciamento estratégico de capacidades”. Ressalta-se que a fábrica não pratica, formalmente, o reuso de código.

Teste

Conforme apresentado no início desta seção, a atividade de teste foi a única descrita com um grau de aprofundamento maior, se comparada com as demais. Em tal atividade, a fábrica de software realiza os testes, unitário e de integração. Já o teste de aceitação é desenvolvido pela unidade contratante da fábrica.

As atividades de teste (unitário e integração) são divididas em metas específicas (ME), e as metas são divididas em práticas específicas (PE), ambas são caracterizadas como tarefas dentro da atividade em questão. A Tabela A.5, apresenta a descrição da meta específica 1 e da prática específica 1.1.

ME1	Tem como objetivo assegurar que as providências para a verificação estejam embutidas nos produtos, requisitos, planos de desenvolvimentos e programas. A verificação inclui a seleção, inspeção, teste e demonstração dos produtos.		
PE 1.1	Descrição	Produtos de trabalhos	Sub-práticas
	Esta prática seleciona os produtos que serão verificados e os métodos usados na verificação de cada produto.	Lista de todos os produtos selecionados para verificação.	Identificar produtos para verificação. Identificar os requisitos para satisfazer o produto selecionado.
		Método de verificação para cada produto selecionado.	Identificar os métodos de verificação que estão disponíveis para o uso.
		Entradas: Produtos que serão verificados Saídas: Lista dos produtos com os métodos de verificação. (a especificação da entrada e saída não estão contempladas no modelo CMMI)	Definir os métodos de verificação para ser usado por cada produto selecionado. Submeter para integração com o plano de projeto a identificação do produto para ser verificado, os requisitos para ser satisfeitos e os métodos para ser usados.
Responsável: <<campo utilizado para definir o papel do responsável>>			

Tabela A.5 - Descrição das metas e práticas específicas que compõem a modelagem da atividade de teste da Fábrica de Software - EMPRESA D.

Ao analisar a tabela acima, é possível perceber que a divisão das atividades do processo em metas e práticas específicas também, podem, ser encontradas no modelo CMMI (CMMI-SE/SW (2002a)). Este fato leva a crer que as demais atividades do processo de produção também seguem a mesma linha, visto que empresa, configurada como objeto desta seção, possui certificação CMMI nível 5.

Em consulta ao modelo CMMI foi possível perceber a ausência das especificações das entradas e saídas inerentes as PE1.1 (vide Tabela A.5). Este fato nos leva a crer que tais informações complementam a PE em questão.

Definida a estrutura geral da atividade de teste, a fábrica de software da EMPRESA D apresentou um dos artefatos utilizados na atividade de teste. Tal artefato pode ser verificado por meio da Figura A.21.

Lista dos produtos para verificação

Identificação do projeto

Produtos Verificados	Tipo do produto	Métodos de Verificação
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Responsável

Figura A.21 - Exemplo de um artefato utilizado na atividade de teste pela fábrica de software (EMPRESA D).

Ao analisar a Figura A.21, é possível verificar que o artefato em questão é utilizado para listar os produtos e os métodos que serão utilizados na verificação (ou teste).

Por fim, é importante salientar que todas atividades pertinentes ao processo de produção, da unidade em questão, utilizam um repositório de boas práticas.

Após a apresentação das atividades operacionais de fábrica e seguindo o roteiro estabelecido na Tabela 6.3, as próximas informações a serem descritas, nesta seção, relacionam-se com modelo de processo utilizado.

Por meio das entrevistas realizadas, foi constatado que a EMPRESA D utiliza, como as EMPRESAS A e B, o modelo de processo incremental, com isto, é válido afirmar que as considerações efetuadas sobre este fato para tais empresas também são válidas para a empresa em questão.

Uma outra constatação, detectada no estudo de caso da EMPRESA D, é a participação dos atores (envolvidos) na execução das atividades do processo, entre eles é possível destacar:

- Área de Treinamento (AT): Responsável pela configuração de todos os treinamentos necessários na fábrica;
- Diretor (DI): Responsável pelo funcionamento da fábrica de acordo com a estrutura global desenvolvida;
- Engenheiro de Software (ES): Responsável pela execução do projeto em si. Pode assumir o papel de analista de sistemas, projetista de software, equalizador, programador ou testador. O engenheiro de software é subordinado ao gerente de projeto. Trabalha, ativamente, no processo de equalização do modelo de negócio e do projeto de software quando estes são desenvolvidos, geograficamente, separados;
- Garantia da qualidade de Software (SQA) – Descrição idêntica à apresentada na seção que relata o caso da EMPRESA A;
- Gerente (G): Responsável por monitorar projetos similares. Cada gerente monitora de 3 a 4 projetos (focados em domínio de conhecimento específico). Cada projeto envolve cerca de 60 pessoas e possui um gerente de projetos;
- Gerente de projeto (GPr): Responsável por monitorar um projeto específico e prestar contas ao gerente;
- Líder Técnico (LT): Responsável pela análise de configuração do ambiente de projeto e pelo monitoramento das principais metodologias e tecnologias para produção de software;

- SEPG: grupo formado pelos gerentes e gerentes de projeto e engenheiros de softwares, com objetivo de definir, acompanhar e avaliar o processo de software;
- Técnico em Configuração (TC) - Descrição idêntica à apresentada nas seções que relata o caso da EMPRESAS A.

É importante salientar que, além dos papéis acima descritos, a EMPRESA D (e não a fábrica de software), possui um líder técnico global e local. O primeiro é responsável por aglutinar questões sobre modelo funcional compartilhado, citado na seção A.4.1, e os aspectos da qualidade de software³⁷. Já o segundo aglutina as questões do sobre modelo funcional compartilhado num âmbito regional, a América Latina, por exemplo.

Uma relação dos atores que compõem a fábrica de software da EMPRESA D com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.22.

Por fim, o processo de produção da fábrica de software da EMPRESA D é caracterizado como pesado, isto é, ele é mais aderente ao *Unified Process* do que ao *eXtreme Program*.

³⁷ Existem unidades que não atingiram certificações de qualidade, como a da fábrica de software visitada. Com isto, a estrutura de liderança global, estabelece prazos e procedimentos para que tais unidades possam atingir os níveis de qualidade exigidos pelo mercado, na qual estas se encontram inseridas. É possível concluir que a estrutura global para o delineamento da qualidade possui várias visões, objetivando, assim, atender possíveis fatores potencializadores, dentro das unidades de produção, e as exigências de mercados continentais e regionais.

Figura A.22 - Distribuição dos envolvidos com o processo de produção da EMPRESA D no modelo de categorização por níveis.

A.4.4 Gestão Operacional

A gestão operacional da fábrica de software da EMPRESA D é dividida em gestão e controle de projetos, gestão da qualidade e gestão de configuração.

Analizada a Figura A.19 e a Figura A.20, é perceptível que a atividade de gestão de projetos possui dois desdobramentos, local e global. No desdobramento local, a fábrica de software, caracterizada como subcontratada, gerencia as atividades de projeto de software (quando configurada com o ciclo médio de produção), equalização, codificação e teste de software (atividades relacionadas ao processo de produção). Para a realização de tal gerenciamento, a fábrica dividiu a atividade em questão nas tarefas de:

- Planejamento de projeto: Tarefa responsável por definir a mão de obra, cronograma e os recursos necessários para o desenvolvimento do projeto,

mapeando riscos relacionados ao desenvolvimento de software. A tarefa de planejamento é desenvolvida com base no tamanho do projeto a ser desenvolvido. Salienta-se que a fábrica de software utiliza o conceito de pontos por função para delinear o tamanho do produto;

- Controle ou acompanhamento de projeto: Na tarefa de acompanhamento de projeto, o gerente de projeto é responsável por verificar se a execução do projeto está de acordo com aquilo que foi planejado.

O desdobramento global da gestão de projetos também possui a mesma divisão encontrada no desdobramento local. A diferença entre as formas de gestão ocorre, apenas, na atividade de execução, pois esta tarefa passa a ser de responsabilidade da fábrica de software, caracterizada aqui como subcontratada de uma unidade de produção ou unidade integradora. A Figura A.23 representa, como é desenvolvido os desdobramentos dentro da EMPRESA D.

Na figura em questão, é possível perceber a presença da porta de negociação de projeto, por ela a unidade de produção negocia junto à fábrica de software a possibilidade de execução de um projeto. Estabelecida a negociação, um contrato de produção é firmado e a atividade de planejamento da fábrica de software é executada. Verifica-se que em tal atividade, a divisão do projeto de software em ordens de serviços, para que estas possam entrar em produção.

Figura A.23 – Configuração da atividade de gestão de projetos (EMPRESA D). Nota: O controle ou acompanhamento global de projeto é desenvolvido remotamente utilizando o conceito de *call conference* (utilização da tecnologia de voz sobre IP em uma conversa remota). No caso apresentado nesta seção as *calls conference* são disparadas pelas unidades localizadas na América do Norte.

A gestão da qualidade, na fábrica de software da EMPRESA D, é realizada sobre duas vertentes: de projeto e de processo.

A gestão da qualidade de projeto está ligada à qualidade do produto gerado. Dados inferidos após a execução da atividade de teste, e número de ordens de serviços corretivas detectadas são utilizados neste tipo de gestão.

Já a gestão da qualidade do processo é de responsabilidade do SQA, desvios dentro da ótica do processo, como por exemplo: falta de padronização no desenvolvimento de artefatos (códigos, formulário) são utilizados para verificar se o processo estabelecido está sendo percorrido corretamente ou não. Reuniões periódicas são realizadas com o objetivo de apresentar o desempenho da equipe de produção da fábrica de software e propor metas da qualidade a serem atingidas.

Por fim, os procedimentos utilizados na atividade de gestão da configuração são semelhantes aos realizados pela fábrica de software da EMPRESA A.

A.4.5 Armazenamento de Dados

Base de Ferramentas

A fábrica de software da EMPRESA D possui, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes às visões gerencial e processual. A relação das ferramentas com as atividades são apresentadas na Tabela A.6.

Atividade do processo		Natureza da ferramenta utilizada para automação da atividade	Visão
Projeto de software		Ferramenta utilizada para desenvolver diagramas relacionados ao projeto de software (caso de uso, classes – paradigma orientado a objetos). Tal ferramenta é, totalmente, integrada com a máquina de processo.	Processual
Equalização		Ferramenta utilizada no desenvolvimento de diagramas utilizados no projeto de software. Software para o desenvolvimento de <i>call conference</i> (vide a definição de <i>call conference</i> na legenda da Figura A.23). Máquina de processo.	
Codificação		Ambiente de desenvolvimento de software para, geralmente configurado em, JAVA e .NET. Controle de versões. Máquina de processo.	
Teste		Máquina de processo. Geradores de casos de testes. Controle de versões. Ambiente de desenvolvimento de software.	
Gestão Operacional	Projetos	Máquina de processo. Gestão de projetos.	Gerencial
	Configuração	Controle de Versões. Máquina de processo.	
	Qualidade	Máquina de processo	

Tabela A.6 – Relação das ferramentas e atividades do processo – EMPRESA D

A máquina de processo, utilizada pela EMPRESA D, possui uma característica que as demais máquinas não possuem, sua estrutura funcional é integrada, por exemplo: no momento que um determinado engenheiro de software está desenvolvendo o projeto, no qual os diagramas que caracterizam a funcionalidade do software são produzidos, a máquina captura, automaticamente, o tempo de tal atividade, armazena as versões geradas, evitando, assim, que o engenheiro se preocupe em abrir a máquina para efetuar o lançamento de tais informações. As palavras proclamadas no exemplo também são válidas para as demais atividades pertinentes ao processo.

Base de Componentes

O reuso de código ou componente, efetuado pela fábrica de software da EMPRESA D, é realizado, informalmente.

A.4.6 Produtos Gerados

A EMPRESA D desenvolve aplicações voltadas aos seguintes domínios de mercado: saúde, manufatura (80% de suas aplicações), telecomunicações, indústria alimentícia e área financeira (bancos). Sabendo que 80% das aplicações da EMPRESA D estão centradas na área de manufatura, tal empresa tende a possuir características de orientação a domínios em seus produtos.

Já em relação à tecnologia, foi verificado que a mesma desenvolve produtos utilizando em várias delas, sendo assim, é possível afirmar que a fábrica de software da EMPRESA D não é orientada à tecnologia. Saliencia-se que o perfil tecnológico para a atuação em um determinado projeto é definido pela área de “gerenciamento estratégico de capacidades”.

Por fim, a orientação a produto traduz a idéia do desenvolvimento e comercialização de um produto apenas, por exemplo: uma empresa qualquer comercializa somente o produto X. Fato este que não ocorre com a EMPRESA D.

A.4.7 Forma de Criação e Organização do Processo

O processo de produção de software desenvolvido pela EMPRESA D foi instituído pela área de “gerenciamento estratégico de capacidades”. A forma de atuação de tal área não foi divulgada pelos entrevistados.

A.5 A Fábrica de Software da EMPRESA E

A.5.1 Caracterização da Empresa

A EMPRESA E foi fundada em 1990 e sua missão está centrada no provimento de soluções, inovadoras e criativas, integrando pessoas e Tecnologia da Informação (TI).

Atualmente, dentro do contexto industrial, a empresa em questão oferece soluções para mais variadas áreas do conhecimento. Entre tais soluções, destacam-se: os sistemas de finanças, de logística, de recursos humanos, de gestão de relacionamento com clientes e de gerenciamento da cadeia de fornecimento. Todos estes sistemas trabalham de forma integrada, sob a ótica conceitual de um ERP. O ERP desenvolvido pela EMPRESA E possui um núcleo, compartilhado por todos seus clientes e customizações são desenvolvidas, com o objetivo, de atender algum requisito específico.

Além da consolidação mercadológica, dentro do contexto de sistemas integrados, a EMPRESA E presta serviços de:

- Treinamento: Atuação efetiva em cada um dos estágios do relacionamento com o cliente, desde a implementação inicial das soluções até a fase final de implantação de seu produto;
- Suporte: Serviços de suporte aos usuários, mantendo sempre um canal aberto para sanar dúvidas, registrar sugestões e fornecer orientações sobre procedimentos técnicos para a utilização do ERP;
- Consultoria: Os serviços de consultoria se estendem a todas as áreas do conhecimento, anteriormente citadas.

Para prover as soluções e prestar os serviços, a empresa em questão conta com cerca de 40 funcionários e está estruturada, organizacionalmente, em:

- Diretoria Geral: Responsável pelas definições estratégicas da empresa, incluindo questões de migração e atualização tecnológica para desenvolvimento dos produtos;
- Área de Recursos Humanos: Responsável pelo gerenciamento dos recursos humanos, imersos nas áreas de administração financeira, projeto de software, produção de software e suporte;
- Área de Administração Financeira: Responsável pelo gerenciamento das questões burocráticas da empresa, como o pagamento de funcionários e recebimento de serviços prestados aos clientes;
- Área de Projeto: Responsável por desenvolver o modelo de negócio e adequar o ERP ao modelo de negócio do cliente (subárea de análise de sistemas). Tal área, também, desenvolve a programação das customizações necessárias para cada cliente;
- Área de Produção: Responsável pela administração da produção, aspecto sobre planejamento e controle da produção são aglutinadas em tal área;
- Área de Suporte: Responsável por sanar as dúvidas geradas com a utilização do ERP, e colher sugestões de melhoria do produto (ou manutenções corretivas) junto ao mercado. Salienta-se que tal área funciona como um termômetro, isto é: se o suporte está sendo muito utilizado por um determinado cliente, pode ser que este necessite de um treinamento, ou o ERP não está atendendo suas necessidades;

O organograma que representa a estruturação da empresa em questão pode ser verificado por meio da Figura A.24.

Figura A.24 - Estrutura organizacional da fábrica de software da EMPRESA E.

Descrita a visão geral da EMPRESA E, o protocolo, apresentado no capítulo 6, estabelece que se relate as impressões sobre os insumos do processo de produção, cujo objetivo é caracterizar a fábrica de software dentro dos ciclos de produção: curto, médio ou longo.

A.5.2 Insumos do Processo

A fábrica de software relatada nesta seção foi concebida sobre o ciclo longo de produção de software. Dentro desta ótica é possível afirmar que tal empresa realiza as atividades de modelagem de negócio, projeto de software (a equalização é classificada como uma tarefa da atividade de projeto de software), produção de código, teste e implantação.

A modelagem de negócio, atividade desenvolvida pelo analista de sistemas, tem como objetivo verificar quais são os ajustes necessários para que o ERP seja customizável à necessidade do cliente. Nesta atividade tais ajustes são materializados e, posteriormente, projetados (atividade de projeto), codificados (atividade de programação), testados (atividade de teste) e, por fim, implantados (atividade de entrega). Uma representação ilustrativa do ciclo de produção estabelecido pela EMPRESA E pode ser verificada por meio da Figura A.25.

Figura A.25 - Modelo de produção da EMPRESA E

Ao analisar a Figura A.25 é possível perceber que as atividades (modelagem de negócio, projetos de software, codificação, teste e implantação) do processo de software tocam as elipses que representam o conceito adaptativo do ERP e as necessidades do cliente. A elipse nuclear não é tocada pelas atividades do processo de produção da fábrica de software em questão, visto que esta parte do ERP se encontra estabilizada.

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção.

A.5.3 Atividades do Processo

Conforme verificado na Figura A.25, o processo de produção de software da EMPRESA E é dividido em: modelagem de negócio, projeto de software, codificação, teste e implantação.

A atividade de modelagem é realizada pelo analista de sistemas, com o objetivo de apresentar as adaptações para que o núcleo do ERP atenda as necessidades dos clientes. Tal modelagem é desenvolvida após uma prospecção de mercado, realizada pela EMPRESA E, com o objetivo de potencializar novos clientes para sua carteira. Quando um novo cliente é prospectado, neste caso o núcleo do ERP já foi apresentado para o cliente e o mesmo estabeleceu um contrato com a

empresa em questão, o analista de sistemas inicia a coleta de requisitos³⁸ para desenvolver o mapa de negócio (caracterizado pela fábrica de software como um artefato) na qual o núcleo de ERP irá operar.

O mapa de negócio tem como objetivo apresentar o contexto sistêmico do cliente e mapear as necessidades adaptativas a serem desenvolvidas junto ao núcleo do ERP. Tal artefato é desenvolvido utilizando as especificações advindas da área de análise de sistemas e validado junto ao cliente.

De posse do mapa de negócio, o analista de sistemas desenvolve, com a participação do programador, o projeto do software. Nesta atividade, é verificado o conceito de equalização presente em vários casos relatados, anteriormente.

Um fato importante deve ser destacado na equalização: o programador (além de discutir as possíveis dúvidas sobre o mapa de negócio) tem contato com as gravações das entrevistas realizadas pelo analista, junto ao cliente prospectado. Fato este que minimiza problemas em relação ao entendimento do mapa de negócio sob a ótica do programador.

Ainda, em relação à equalização, os entrevistados disseram que, periodicamente, alguns programadores participam, juntamente com o analista de sistemas, da modelagem de negócio e do projeto de software. Fato este que estreita as relações entre o programador e o analista, pois o primeiro tem a oportunidade de conhecer a realidade de trabalho do segundo³⁹.

De posse das informações acima é perceptível que a atividade de modelagem de negócio (ou análise de sistemas) e projeto de software se relacionam, ativamente, com os seguintes artefatos:

³⁸ No processo de coleta de requisitos desenvolvido pela EMPRESA E, o analista de sistemas utiliza técnicas de simulação de cenários e entrevistas com o cliente, com o objetivo de verificar quais são as necessidades adaptativas inerentes ao núcleo do ERP.

³⁹ É importante salientar que no plano de carreira, estabelecido pela empresa em questão, todos os analistas conhecem a realidade dos programadores, pois estes, antes de estarem atuando como analistas foram programadores dentro do contexto da EMPRESA E.

- Mapa de negócio: Artefato resultante da modelagem de negócio e utilizado na atividade de projeto de software;
- Gravações e questionários gerados na coleta de requisitos: Utilizados no desenvolvimento do mapa de negócio e utilizados pelos analistas e programadores nos procedimentos de equalização;
- Documento textual utilizado para materializar os eventos do modelo de negócio e as funcionalidades do projeto de software.

Materializadas as funcionalidades adaptativas do ERP, a atividade de codificação é executada. Conceitos relacionados ao reuso de código são utilizados, formalmente, em tal atividade. A empresa em questão possui uma biblioteca de componentes, na qual armazena o núcleo do ERP. Os componentes são classificados como genéricos (componentes que possuem um alto grau de reutilização – componente de infra-estrutura) ou específicos (componentes que possuem um baixo grau de reutilização).

Realizada a atividade de codificação, as funcionalidades são testadas utilizando as técnicas de caixa preta e de caixa branca (SOMMERVILLE (2003)).

Além de tais testes, a empresa em questão utiliza o conceito de testes de imersão (termo utilizado pelos envolvidos com o processo de produção no momento das entrevistas). Um teste de imersão tem como objetivo testar um determinado software em um ambiente real de operação com dados reais. Tal teste está dividido em:

- Alfa teste: As adaptações geradas para um determinado cliente são testadas utilizando os dados produzidos pelo ERP que opera na própria fábrica⁴⁰.

⁴⁰ Salienta-se que a EMPRESA E utiliza o próprio ERP para automatizar o seu sistema de informação gerencial.

Ressalta-se que para a realização do teste em questão, a fábrica de software espelha o ambiente do cliente em um laboratório de teste;

- Beta teste: As adaptações geradas para um determinado cliente são testadas utilizando os dados produzidos pelo ERP que opera em uma outra empresa. Para realização deste teste, a fábrica de software, solicita a uma empresa que opera com o seu ERP, dados para a realização dos testes das customizações do cliente em questão;
- Gama teste: Teste de aceitação, na qual as funcionalidades são testadas na presença do cliente. O teste em questão é desenvolvido dentro da atividade de implantação.

A atividade de implantação tem como objetivo implantar o ERP no cliente prospectado. Nesta atividade são desenvolvidos os treinamentos e tarefas relacionadas às manutenções corretivas que, quando detectadas, são realizadas. Salienta-se que após a implantação, o cliente recebe todo o suporte para a utilização do produto, com isto, dúvidas são esclarecidas e melhorias são detectadas.

Por fim, a atividade de revisão⁴¹, apresentada no capítulo 3, é praticada pela EMPRESA E. Esta atividade é praticada sempre que o núcleo do ERP é alterado, tecnologicamente e funcionalmente. Tal núcleo é disponibilizado, indistintamente, para todos os clientes sem que estes solicitem, caracterizando, assim, a prática da atividade em questão.

Após a apresentação das atividades operacionais da fábrica, e seguindo o roteiro estabelecido na Tabela 6.3, as próximas informações a serem descritas neste texto relacionam-se com modelo de processo utilizado pela fábrica software em questão.

⁴¹ Atividade essa que tem, como objetivo, trocar o componente de um software que está implantado, por um outro, que provenha maior otimização (por exemplo: relação a tempo de reposta).

Por meio das entrevistas realizadas, foi constatado que a empresa utiliza, como as demais, o modelo de processo incremental.

Uma outra constatação detectada é a participação dos atores (envolvidos) na execução das atividades do processo. Na descrição das atividades processuais e gerencias, é possível verificar a presença de:

- Analista de Sistemas (AS): Responsável por colher os requisitos, desenvolver a modelagem de negócio e o projeto de software. O analista deve ter amplos conhecimentos sobre as técnicas de levantamento de requisitos, modelagem de negócio utilizando os paradigmas existentes (atualmente, se fala em orientação a processos e orientação a objetos);
- Diretor Geral (DG): Responsável pelo funcionamento da fábrica de acordo com a estrutura global desenvolvida. Prospecta e potencializa possíveis clientes. Direciona questões relacionadas à migração tecnológica dentro do ambiente de produção. Define as políticas estratégicas da organização;
- Equipe de Suporte (ESp): Responsável por sanar as possíveis dúvidas sobre a utilização do produto e colher sugestões do cliente;
- Grupo de Garantia da Qualidade (GGQ): é um grupo formado pelos analistas de sistemas, e programadores, com objetivo de definir, acompanhar e avaliar o processo de software e verificar a qualidade do produto;
- Programador (P): Responsável pela codificação das customizações funcionais mapeadas na atividade de projeto de software. Participa, ativamente, da equalização;
- Programador de Produção (PP): Atua como um gerente de projetos, cuja responsabilidade é planejar aspectos relacionados à produtividade, monitorar projetos executados e estabelecer métricas a serem cumpridas. A garantia da

qualidade dos produtos gerados, também, está na alçada das responsabilidades do programador de produção;

- Técnico em Configuração (TC) - Descrição idêntica à apresentada na seção que relata o caso da EMPRESAS A.

Uma relação dos atores que compõem a fábrica de software da EMPRESA E com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.26.

Figura A.26 - Distribuição dos envolvidos com o processo de produção da EMPRESA E no modelo de categorização por níveis.

Por fim, o processo de produção da fábrica de software da EMPRESA E utiliza alguns conceitos e técnicas direcionadas pelo XP, tais como programação em pares e o cartão de histórias.

A.5.4 Gestão Operacional

As atividades relacionadas à gestão operacional da EMPRESA E são: gerenciamento de projetos, de configuração e controle da qualidade.

A atividade de gerenciamento de projeto é dividida em 3 tarefas, planejamento, execução e controle.

A tarefa de planejamento é responsável por estabelecer cronogramas, recursos e mapear os custos para a execução de um determinado projeto de software. Para a realização desta tarefa, o programador de produção, divide o projeto arquitetônico do software em artefatos gerados pela atividade de projeto de software, aglutinando-os em ordens de serviço.

As ordens de serviços possuem os seguintes atributos:

- Nome do Projeto;
- Programador responsável pelo seu desenvolvimento;
- Versão da unidade de software⁴² a ser desenvolvida;
- Prioridade para desenvolvimento: Uma unidade de software, em relação ao seu desenvolvimento, pode ser classificada como: prioritária, normal ou de baixa prioridade. Salienta-se que as unidades prioritárias devem ser desenvolvidas com maior eficiência em relação ao tempo;
- Tempo estimado para o desenvolvimento da unidade de software;
- Tempo utilizado para o desenvolvimento;

⁴² Ressalta-se que uma unidade de software é gerada a partir de uma funcionalidade e uma funcionalidade é gerada a partir de um evento sistêmico.

- Histórico do desenvolvimento da unidade de software: Atributo responsável por armazenar todas as informações relacionadas à unidade de software presente na ordem de serviço. A fábrica de software em questão utiliza um documento que aglutina todas as informações relacionadas à unidade de software;
- Origem da ordem de serviço: Uma ordem de serviço pode ser gerada a partir de um evento sistêmico e, conseqüentemente, por uma funcionalidade mapeada na atividade de projeto de software, ou por meio de uma solicitação de correção ou melhoria capturada pela área de suporte.

As informações capturadas pela ordem de serviço são armazenadas no sistema de planejamento e controle da produção⁴³ (PCP), que se configura como a máquina de processo da empresa em questão, para, posteriormente, serem executadas.

A execução de uma OS está, intimamente, ligada ao desenvolvimento das atividades de codificação e teste. Neste momento, os envolvidos com tais atividades devem apontar, sobre a ordem de serviço, as horas trabalhadas e o percentual de desenvolvimento da unidade de software. Estas informações são analisadas pelo programador de produção, com a finalidade de se estabelecer a tarefa de controle, relacionada ao conceito de gestão de projetos. Ressalta-se que tal tarefa tem, como finalidade, verificar o andamento do projeto e, se necessário for, replanejar os aspectos relacionados a cronograma, recursos e custos de desenvolvimento.

Por fim, a fábrica de software em questão utiliza uma base histórica de projetos para realizar as estimativas, amplamente, utilizadas na tarefa de planejamento de projeto.

⁴³ O sistema de planejamento e controle da produção da empresa em questão também é utilizado por outros 17 clientes que prestam serviços nos mais variados domínios do conhecimento.

qualidade podem verificar o motivo que levou um determinado ator a cometer tal desvio.

Por fim, os procedimentos utilizados na atividade de gestão da configuração são semelhantes ao realizado pela fábrica de software da EMPRESA A.

A.5.5 Armazenamento de Dados

Base de Ferramentas

A fábrica de software da EMPRESA E possui, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão gerencial e processual. A relação das ferramentas com as atividade são apresentadas na Tabela A.7.

Atividade do processo		Natureza da ferramenta utilizada para automação da atividade.	Visão
Modelagem de negócio		Editor de texto. Máquina de processo.	Processual
Projeto de software		Editor de texto. Máquina de processo.	
Codificação		Editor de texto. Máquina de processo. Ambiente de desenvolvimento de código. Controle de versões.	
Teste		Editor de texto. Máquina de processo. Ambiente de desenvolvimento de código. Controle de versões.	
Implantação		Editor de texto. Máquina de processo. Ambiente de desenvolvimento de código.	
Gestão operacional	Projetos	Máquina de processo. Gestão de projetos.	Gerencial
	Configuração	Controle de Versões. Máquina de processo.	
	Qualidade	Máquina de processo.	

Tabela A.7 – Relação das ferramentas e atividades do processo – EMPRESA E. Nota: O Editor de texto é utilizado para verificar a especificação de uma funcionalidade a ser implementada em uma unidade de software.

As mesmas considerações, sobre a máquina de processo, efetuada para a fábrica de software da EMPRESA A, também, são válidas para a fábrica de software apresentada nesta seção.

Base de Componentes

O reuso de código e de componentes é, amplamente, praticado pela EMPRESA E, principalmente, no que diz respeito ao núcleo do produto por ela comercializado, o ERP.

A.5.6 Produtos Gerados

Na seção sobre a caracterização da empresa, foi possível perceber que a mesma comercializa vários sistemas integrados em um ERP. Em entrevista junto aos envolvidos com o processo, foi questionada se a mesma dá preferência para clientes que desejam somente este tipo de produto. A resposta foi positiva. Sendo assim, é possível afirmar que EMPRESA E é orientada a domínios e organiza sua produção em células para atender clientes específicos.

Já em relação à tecnologia, foi verificado que a fábrica desenvolve produtos utilizando uma única (tecnologia Microsoft), sendo assim é possível afirmar que a fábrica de software da EMPRESA E é orientada à tecnologia.

Por fim, a orientação a produto, traduz a idéia do desenvolvimento e comercialização de um produto apenas. Fato este que ocorre com a EMPRESA E.

A.5.7 Forma de Criação e Organização do Processo

O processo de produção de software da empresa em questão foi configurado a partir das experiências dos profissionais que nela atuam.

A.6 A Fábrica de Software da EMPRESA F

A.6.1 Caracterização da Empresa

A EMPRESA F atua, desde 1986, nos setores da tecnologia e informática. Com aplicações direcionadas para as áreas de comunicação de dados, internet, desenvolvimento de sistemas e novas tecnologias, consultoria, suporte técnico e alocação de recursos humanos especializados e experiência em integração de plataformas heterogêneas, conduz projetos desde o desenvolvimento até os serviços de suporte e operação.

A empresa em questão foca, diretamente, o aperfeiçoamento do seu processo de terceirização, provendo soluções relacionadas a programas internos de treinamento e acompanhamento de seus colaboradores, alocados em clientes ou não. A prestação de serviços de consultoria e desenvolvimento de projetos na área de informática é um dos objetivos imersos no contexto da empresa.

Atualmente, a carteira de clientes da EMPRESA F, conta com instituições ligada ao: mercado financeiro (bancos, seguradoras, administradoras de cartões de crédito, financeiras, corretoras e assessores de investimento); comércio (redes de postos de gasolina, restaurantes, lojas); setores de produção e serviços (distribuidores, revendas, varejo, atacado e indústrias); área de auditoria e inspetoria; área de atendimento a clientes e laboratórios farmacêuticos.

Os principais produtos da empresa em questão são sistemas de captura e consulta de cheques, assinaturas e autenticação de pagamentos eletrônicos, trocas de arquivos via internet, compras e vendas de ações, soluções de computação móvel, sistemas eletrônicos de compra via Internet e *help desk*.

A EMPRESA F opera com 5 núcleos de trabalho, quatro localizados na grande São Paulo e um localizado em Brasília. Cerca de 600 colaboradores trabalham, diretamente, ligados à empresa.

Estruturalmente, a EMPRESA F mantém, sob a diretoria de negócios, diversos grupos de trabalho, dos quais se destacam:

- A divisão de produto: Divisão encarregada de analisar e parametrizar as soluções comercializadas, definindo viabilidades e especificidades necessárias à sua produção. Analisa, também, mercados e tendências em relação a produtos, a fim de determinar estratégias (ou administrar as já implantadas) de rastreamentos de aceitabilidades e replicabilidades de produtos, fomentando a escalabilidade comercial e produtiva;
- A divisão de tecnologia: Divisão encarregada de providenciar condições para a produtividade e o atendimento das contratações e das tendências percebidas de mercado, pesquisando novos paradigmas em tecnologias, ferramentas e comunicabilidade, providenciando cursos e treinamentos, estruturando e mantendo a fabricação do software e garantindo-lhe padrões da qualidade;
- A divisão de consultoria: Divisão encarregada da pesquisa necessária em tecnologias e ferramentas, para o desenvolvimento de novas soluções e novas arquiteturas e atua nas discussões sobre melhoria contínua e domínio sobre o conhecimento.
- Divisão de Gerência de Projetos: O grupo de projetos incumbe-se da arte necessária ao desenvolvimento de interfaces para os produtos a construir, além da responsabilidade para com toda a estrutura de comunicação, via TI, da empresa.

A Figura A.28 apresenta o organograma que reflete a estrutura organizacional da empresa em questão.

Figura A.28 – Estrutura Organizacional da EMPRESA F

Descrita a visão geral da EMPRESA F, o protocolo, descrito no capítulo 6, estabelece que se relate as impressões sobre os insumos do processo de produção, cujo objetivo é caracterizar a fábrica de software dentro dos ciclos de produção: curto, médio ou longo.

A.6.2 Insumos do Processo

A EMPRESA F divide a produção de software entre a fábrica de projetos e a fábrica de software.

A fábrica de projeto é responsável por realizar o levantamento dos requisitos, a análise de sistemas, o projeto de software, a entrega e a manutenção. Como fábrica de software, a empresa denomina a produção de código com seu detalhamento, ou seja, a implementação propriamente dita, envolvendo equalização das ordens de serviços, codificação e testes unitários.

Dentro da ótica do processo e do produto, a empresa está estruturada em grupos de trabalho. O grupo da qualidade encarrega-se de definir métodos e critérios para medir e garantir a qualidade, analisando especificidades e conformidades entre requisitos e componentes produzidos. O grupo de teste define a tecnologia necessária e planeja a prática deste tipo de atividade, para

experimental o que se produz em matéria de componentes, definindo planos e baterias de testes, além de aplicá-los nas integrações e no produto final. O grupo administrador da rede incumbe-se em disponibilizar toda a tecnologia necessária aos outros grupos, tornando funcional o complexo tecnológico: equipamentos, sistemas operacionais, estruturas de comunicação, redes físicas, aplicativos e artefatos diversos. O grupo de cursos técnicos encarrega-se de gerir talentos e carreiras, fomentando oportunidades de disseminação de conhecimento através de eventos como cursos, treinamentos e workshops, entre outros.

A execução do ciclo longo é de responsabilidade dos analistas de negócio e dos projetistas de software. O analista desenvolve o levantamento de requisitos, mapeia os eventos sistêmicos e desenvolve o modelo de negócio. Já o projetista, de posse de tal modelo, desenvolve o projeto de software. Tal projeto é “fatiado” em funcionalidades, as quais estas irão compor as ordens de serviços que serão executadas pela fábrica de software (execução do ciclo curto). Realizado o teste de integração, a fábrica de projetos recebe os produtos, inerentes às ordens de serviços, e os implanta no cliente. Uma visão ilustrativa da ciclo de produção praticado pela EMPRESA F pode ser verificado por meio da Figura A.29.

Figura A.29 – O Ciclo de Produção praticado pela EMPRESA F. Fn = Funcionalidade. OS = Ordem de Serviço.

De posse dos ciclos de produção, e com base no protocolo, descrito no capítulo 6, a próxima seção irá apresentar as informações inerentes às atividades do processo de produção.

A.6.3 Atividades do Processo

Conforme verificado na Figura A.29, as atividades inerentes ao processo de produção de software da unidade fabril, imersa no contexto organizacional da EMPRESA F, são: levantamento de requisitos, análise de sistemas, projeto de software, equalização, codificação, teste, entrega e manutenção. A fábrica de software, unidade visitada para realização deste estudo, executa apenas as atividades de equalização, codificação e testes.

A atividade de equalização tem como objetivo verificar se as ordens de serviços, geradas pela atividade de projeto de software, estão completas e corretas para que estas possam ser colocadas em linha de produção. Para desenvolver a atividade de equalização é necessário executar os seguintes passos:

- Verificar se as ordens de serviços geradas respeitam o padrão documental estabelecido pelo processo de produção de software;
- Verificar se as ordens de serviços geradas possuem todos os artefatos necessários para que a mesma possa ser colocada em linha de produção;
- Verificar se os artefatos, inerentes às ordens de serviços, estão consistentes em relação à sua concepção conceitual. Por exemplo: O diagrama de entidades e relacionamentos, deve espelhar as cardinalidades pertencentes às tabelas⁴⁴;

⁴⁴ O exemplo exposto no texto foi apresentado pelo entrevistado.

- Aprovar ou rejeitar a ordem de serviço. A aprovação de uma ordem de serviço está, intimamente, ligada às verificações descritas, anteriormente. Caso a ordem de serviço não atenda os critérios de verificação a mesma é rejeitada;

Após passar pela equalização, a OS entra em linha de produção, neste momento as funcionalidades são codificadas e testadas. Ressalta-se que na atividade de programação o reuso de código é praticado informalmente. Já a atividade de teste se divide em:

- Teste unitário: Testes de caixa preta e caixa branca são executados, em cada funcionalidade, pela célula de teste (grupo de pessoas responsável por testar as ordens de serviços geradas);
- Teste integrado: Teste de caixa preta é executado no âmbito da OS. Lembrando sempre, que neste caso, a OS integra várias funcionalidades.

Quando algum erro é encontrado em alguma funcionalidade, a OS retorna às mãos do programador, para que as correções sejam efetuadas. Os erros são apontados na máquina de processo e estes servem de base para verificar a qualidade do produto desenvolvido pela fábrica de software.

A EMPRESA F possui um banco de testes que é alimentado, constantemente, pelos projetos que estão em desenvolvimento. A alimentação de tal banco é de responsabilidade das pessoas envolvidas com a célula de teste. Estruturalmente, o banco está dividido por domínio da aplicação e por projeto. Por exemplo: o domínio “software da indústria farmacêutica” possui os projetos *A* e *B*.

Todos os envolvidos com o processo de produção da fábrica de software em questão devem realizar o apontamento de horas trabalhadas na máquina de processo. Tal apontamento será utilizado pelo líder de célula para o acompanhamento do projeto.

Por fim, o *checklist* para o desenvolvimento dos testes e o documento de padronização de códigos são alguns artefatos utilizados pela fábrica de software em questão. Salienta-se que a EMPRESA F não permitiu que as informações contidas nos artefatos fossem divulgadas neste trabalho.

Após a apresentação das atividades operacionais da fábrica e seguindo o roteiro estabelecido na Tabela 6.3, as próximas informações a serem descritas neste texto relacionam-se com modelo de processo.

A fábrica de software da EMPRESA F utiliza uma mistura do modelo evolucionário, focado na técnica de prototipação, e do incremental. Neste caso, o analista de negócio colhe os principais requisitos, desenvolve a análise sistêmica, mapeando os principais eventos, projeta as funcionalidades relacionadas aos requisitos levantados, configura uma OS solicitando o desenvolvimento do protótipo (caracterizado pela empresa como protótipo de interface) e repassa tal ordem para fábrica de software. A fábrica desenvolve o protótipo de interface e o entrega ao analista de negócio ou ao projetista de software. Estes, por sua vez, apresentam tal protótipo ao cliente, objetivando colher novos requisitos que espelhem novas funcionalidades. As funcionalidades são aglutinadas, utilizando o modelo incremental, nas ordens de serviço e colocadas em linha de produção. Tais funcionalidades são produzidas e, novamente, atingem o cliente. Ressalta-se que neste momento, o protótipo de interface, começa a ser encarado como um produto.

Uma outra constatação detectada é a participação dos atores (envolvidos) na execução das atividades do processo. Na descrição das atividades processuais e gerencias, é possível verificar a presença do:

- Analista e Negócio (AN) e Projetista de Software (PR): Já definidos anteriormente.
- Célula de Testes (CT) – grupo de programadores que tem a responsabilidade de apoiar a realização dos testes funcionais;

- Diretor de Negócio (DN) – é um papel gerencial, com a responsabilidade de potencializar novos clientes dentro dos domínios de conhecimento nos quais a fábrica de software atua; trabalha, diretamente, no desenvolvimento das estratégias relacionadas ao negócio da empresa;
- Diretor de Tecnologia (DT) – é um papel gerencial, com a responsabilidade de potencializar novas tecnologias para o desenvolvimento de novos produtos. Responsável por coordenar, taticamente, os demais envolvidos com estrutura organizacional da fábrica de software;
- Garantia da Qualidade de Software (SQA) – Descrição idêntica à apresentada na seção que relata o caso da EMPRESA A;
- Gerente de Produção (GP) – é um papel gerencial dentro da equipe, com as responsabilidades de coordenar as equipes de equalização, de implementação e de testes dos projetos;
- Grupo de Processo e Engenharia de Software (SEPG) – Descrição idêntica à apresentada na seção que relata o caso da EMPRESA A.
- Líder de Célula (LC) – é um papel operacional dentro da equipe, com a responsabilidade de assessorar a coordenação da produção em atividades de acompanhamento de projetos e orientação técnica da equipe de produção;
- Programador (P) - é um papel operacional e direcionado para o domínio de técnicas de programação e ferramentas de desenvolvimento, com a responsabilidade de equalizar, implementar e testar as funcionalidades inerentes às ordens de serviços;
- Técnico em Configuração (TC) – Descrição idêntica à apresentada na seção que relata o caso da EMPRESA A;

Uma relação dos atores que compõem a fábrica de software da EMPRESA F com o modelo composto de categorização por níveis, apresentado por TRINDADE (2006), pode ser verificado na Figura A.30.

Figura A.30 - Distribuição dos envolvidos com o processo de produção da EMPRESA F no modelo de categorização por níveis.

Por fim, o processo de produção da fábrica de software da EMPRESA F é caracterizado como pesado, isto é, ele é mais aderente ao *Unified Process* do que ao *eXtreme Program*.

A.6.4 Gestão Operacional

A gestão operacional da fábrica de software em questão é dividida em: gestão de projetos; gestão de configuração e; gestão da qualidade.

A atividade de gestão de projetos é responsável pelo planejamento e controle da execução do projeto. De posse das ordens de serviços, geradas na atividade de

projeto de software, o gerente de produção consulta a base histórica⁴⁵, verificando a existência de um projeto semelhante, para desenvolver o cronograma, os recursos necessários para desenvolvimento do produto. Ressalta-se, também, que o conceito de pontos por função é utilizado nas estimativas aludidas na fase de planejamento de projeto. O principal artefato gerado pela atividade em questão é o plano para a execução do projeto. Nele estão contidas todas as informações relacionadas às estimativas aludidas pelo gerente de produção.

Desenvolvido o planejamento, o projeto entra na fase de execução, com isto, as atividades do processo de produção são executadas. Por fim, a fase de controle tem como objetivo verificar se o que foi planejamento está sendo executado, se desvios forem detectados o plano de projeto é alterado.

Por fim, as atividades de gerenciamento de configuração e da qualidade são semelhantes às realizadas pela fábrica de software da EMPRESA A.

A.6.5 Armazenamento de Dados

Base de Ferramentas

A fábrica de software da EMPRESA F possui, em sua estrutura, um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes à visão gerencial e processual. A relação das ferramentas com as atividades é apresentada na Tabela A.8.

⁴⁵ As informações contidas na base histórica de projetos são geradas pela máquina de processo.

Atividade do Processo		Natureza da ferramenta utilizada para automação da atividade.	Visão
Análise de sistemas		Não praticada pela fábrica.	Processual
Projeto de software			
Equalização		Máquina de processo. Controle de versões. Ferramenta para visualização dos artefatos advindos da atividade de projeto de software. Gestão de projetos.	
Codificação		Ambiente de desenvolvimento de software. Software para controle de versões. Máquina de processo.	
Teste		Máquina de Processo. Controle de versões. Ambiente de desenvolvimento de software	
Implantação		Não praticada pela fábrica.	
Entrega			
Gestão operacional	Projetos	Máquina de processo. Gestão de projetos.	Gerencial
	Configuração	Controle de Versões. Máquina de processo.	
	Qualidade	Máquina de processo.	

Tabela A.8 – Relação das ferramentas e atividades do processo – EMPRESA F.

As mesmas considerações, sobre a máquina de processo, efetuadas para a fábrica de software da EMPRESA A, também, são válidas para a fábrica de software apresentada nesta seção.

Base de Componentes

O reuso de código ou componente, efetuado pela fábrica de código da EMPRESA F, é realizado, informalmente.

A.6.6 Produtos Gerados

Na seção sobre a caracterização da empresa, foi possível perceber que a mesma comercializa vários tipos de sistemas. Em entrevista junto aos envolvidos com o processo, foi questionado se a mesma dá preferência para algum domínio de aplicação. A resposta obtida foi a seguinte:

“A empresa desenvolverá produtos que possam ser, amplamente, replicados em outros clientes”. (informação obtida nas entrevistas junto aos membros da empresa)

De posse desta informação, é possível afirmar que a fábrica de software em questão não tende a ser orientada a produtos.

Já em relação à tecnologia, foi verificada que a fábrica desenvolve produtos utilizando várias, sendo assim é possível afirmar que a fábrica de software da EMPRESA F não é orientada à tecnologia.

Por fim, a orientação a produto, traduz a idéia do desenvolvimento e comercialização de um ou dois produtos apenas, por exemplo: uma empresa qualquer comercializa somente o produto X. Fato este que não ocorre com a EMPRESA F.

A.6.7 Forma de Criação e Organização do Processo

As considerações, sobre a forma de criação e organização do processo de produção de software, efetuados no estudo de caso da EMPRESA F são semelhantes as da EMPRESA A.

Anexo C – Requisitos Funcionais da Aplicação a ser Codificada e Testada pela Fábrica de Software da FATEC

Este anexo tem como objetivo apresentar a especificação dos requisitos funcionais do projeto de software a ser implementado pela fábrica de software da Faculdade de Tecnologia de Ourinhos (FATEC-OU). Segue a lista dos requisitos:

- **Cadastro de Currículo:** O cadastro de currículo permite ao candidato, preencher seu currículo através de um formulário, disponibilizado no portal corporativo. Tal funcionalidade conta com os seguintes métodos:
 - Inclusão: Funcionalidade responsável pela inclusão de currículos pelos candidatos;
 - Atualização: Funcionalidade responsável pela atualização de currículos pelos candidatos;
- **Cadastro de Formação Escolar:** O cadastro de Formação mantém informações da formação escolar dos profissionais, e da formação exigida pelas funções da organização. Tal funcionalidade, como todas as apresentadas adiante, conta com os seguintes métodos:
 - Inclusão: Funcionalidade responsável pela inclusão de Formação Escolar;
 - Alteração: Funcionalidade responsável pela alteração de Formação Escolar;
 - Exclusão: Funcionalidade responsável pela exclusão de Formação Escolar.

- **Cadastro de Treinamento:** O cadastro de Treinamento mantém informações de treinamentos dos profissionais, e dos treinamentos exigidos pelas funções da organização. Permite indicar se o treinamento é de responsabilidade da organização e os tipos diferenciados de treinamento: *on the job training*, *coaching*, *mentoring*, *e-learning*, auto-estudo;
- **Cadastro de Habilidades:** O cadastro de habilidades mantém informações das habilidades dos profissionais, e das habilidades exigidas pelas funções da organização;
- **Cadastro de Conhecimentos:** O cadastro de conhecimentos mantém informações dos conhecimentos dos profissionais, e dos conhecimentos exigidos pelas funções da organização. O cadastro de conhecimentos permite indicar o tipo e o nível de conhecimento;
- **Cadastro de Funções Desempenhadas:** O cadastro mantém informações das funções desempenhadas na organização. Cada função permite indicar a formação escolar, treinamento, habilidades e os conhecimentos exigidos;
- **Cadastro de Certificações:** O cadastro de certificações mantém informações das certificações dos profissionais. O cadastro de certificações permite indicar a instituição e o exame realizado;
- **Cadastro de Profissionais:** O cadastro de profissionais permite incluir, alterar e excluir os seguintes dados da ficha do profissional: dados pessoais; dados profissionais; local de trabalho; currículo; formação; treinamentos realizados; habilidades; conhecimentos; certificados; laudo técnico; experiência; cargo; funções desempenhadas na organização.
- **Módulo aglutinador de Profissionais:** Módulo responsável pelo cadastro e manutenção dos dados dos profissionais. Tal módulo deve estar dividido em:

- Candidatos (quem enviou currículo);
 - Candidatos aprovados pelo RH (quem já passou pelo teste psicológico);
 - Candidatos aprovados em entrevista técnica;
 - *Free lancers*;
 - Colaboradores¹;
 - Ex-Colaboradores.
- **Menus:** Os menus de acesso às funcionalidades do software devem ser apresentados de acordo com o perfil do usuário. Salienta-se que o software deve possuir os seguintes menus:
 - Manutenção Cadastros: Menu que dá acesso às páginas de funcionalidades comuns do software;
 - Manutenção Profissionais: Menu que dá acesso à página de manutenção de profissionais;
 - Consulta Profissionais: Menu que dá acesso à página de consulta de profissionais;
 - Acesso às funcionalidades do software: Funcionalidade responsável pela criação dos menus de acordo com o perfil do usuário.

¹ Um colaborador possui apenas um cargo, mas pode desempenhar mais de uma função na organização, serão reportadas as divergências em relação às competências necessárias para a função desempenhada. Na ficha do colaborador será registrado o *login* de rede e perfil de acesso, de modo a integrar a autenticação de controle de acesso. Nota: Os usuários sem perfil de acesso definido não terão acesso ao sistema.

Além dos requisitos funcionais a ERS contempla os requisitos não funcionais, entre eles é possível destacar:

- Requisitos do Ambiente Operacional. O software deve acessar o Banco de dados *MySQL 4.1* via Servidor *Web Apache 2.0*. A linguagem escolhida para codificação do produto é o *script PHP*;
- Interface: Padrão do portal corporativo da EMPRESA X;
- Codificação: A codificação deve respeitar o *Java Code Conventions*;
- Segurança: Senha do usuário criptografada na base de dados e *string* de conexão da base de dados criptografada;
- Treinamento: Apresentação e treinamento do software para um grupo de usuários;
- Manuais: Documento com telas e texto explicativo sobre como operar o software;
- Critérios de Aceitação: Aprovação do protótipo dos módulos do software; Homologação em ambiente de teste.

Anexo B – Relato Detalhado das Fábricas Japonesas e Americana

Este anexo tem como objetivo apresentar a descrição detalhada do caso de cinco empresas, quatro delas situadas no Japão (Hitachi, Toshiba, NEC e Fujitsu) e uma nos Estados Unidos (System Development Corporation).

Salienta-se que as informações colhidas no referencial bibliográfico utilizado para o desenvolvimento deste anexo estendem as informações apresentadas na seção 2.1 deste trabalho.

B.1 Caracterização das Empresas

Antes de iniciar a caracterização das empresas que compõem este anexo, é válido afirmar que o texto de CUSUMANO (1991), não apresenta organograma e uma seção que relate a missão das empresas.

B.1.1 Caracterização da SDC

A SDC é uma empresa sem fins lucrativos, financiada pelo governo americano, e nos meados de 1950, foi contratada pela força aérea para o desenvolvimento de um sistema de controle e de comando da defesa do espaço aéreo.

Em 1960, a SDC continuou prestando serviço para o departamento de defesa sendo que os principais projetos desenvolvidos nesta época foram: controladores de satélites, vários tipos de simuladores e a participação na missão espacial Apollo¹.

¹ Na década de 1960 a SDC, também, desenvolveu o sistema operacional de tempo compartilhado denominado como Q-32. Tal sistema era executado no *mainframe* AN/FSQ-32 utilizado pela rede ARPA (rede esta que deu origem a Internet). O Q-32 foi o primeiro sistema a suportar múltiplos usuários e a comunicação de computadores.

Para o setor privado a empresa desenvolveu softwares para os seguintes domínios: gerenciamento hospitalar, bibliotecas e aeroportos.

Por volta de 1969, a SDC já era uma organização sólida no mercado americano e abandonou o *status* de empresa sem fins lucrativos e passou a competir, fortemente, por contratos no âmbito internacional. A forte concorrência, principalmente, com as empresas japonesas levou a SDC a enfrentar dificuldades financeiras. Os fatores que contribuíram para este fato foram:

- Inexperiência para competir por contratos advindos do governo americano, pois até 1969, a SDC não necessitava participar de licitações e concorrências;
- O ingresso no mercado privado de produção de software levou a SDC a concorrer, diretamente, com as grandes empresas produtoras de software da época;
- Necessidade de escrever software para outras arquiteturas de computadores e não mais para uma arquitetura única, operante dentro do contexto militar aeronáutico americano;
- E, por fim, ao deixar de ser uma empresa sem fins lucrativos a SDC não mais obteve incentivos e subsídios advindos do governo.

As dificuldades financeiras levaram a SDC a um declínio no número de funcionários, em 1963, a empresa possuía 4.300, já em 1969, a empresa passou a ter 3.200 e, em 1971, apenas 2.000.

Os problemas enfrentados levaram a SDC a mudar a sua política estratégica em relação ao domínio de atuação, a empresa passou a atuar em outros domínios de mercado com maior impacto, tais como: o controle de tráfego aéreo e o comando e controle inteligente para o departamento de polícia. Com o desenvolvimento de tais produtos a empresa obteve um alto grau de “replicabilidade” de produtos,

dobrando, assim, a sua arrecadação de 45 milhões de dólares, em 1971, para 90 milhões de dólares, em 1974.

Em 1980, a SDC foi vendida para a *Burroughs Corporation* e esta, por sua vez, se fundiu com a *Sperry Corporation* em 1986, e formaram a *Unisys*. Este fato levou a SDC a se transformar na *Paramax*, caracterizada como uma subsidiária da *Unisys*. Em 1995, a *Paramax* foi vendida para *Loral Corporation* e no ano seguinte a *Loral* vendeu-a para *Lockheed Martin*.

B.1.2 Caracterização da Hitachi

CUSUMANO (1991) relata que a Hitachi iniciou suas atividades em 1908. Por volta de 1957, a empresa iniciou a sua produção de computadores e softwares. Ressalta-se que a empresa concentrava-se no desenvolvimento de sistemas de informações focando o domínio bancário, telefonia e governo.

Em 1969, dois fatores contribuíram para que a Hitachi instalasse sua fábrica de software:

- escassez de mão de obra qualificada (na época programadores) e, principalmente;
- repetir o sucesso do conceito de manufatura já alcançado com a produção de componentes eletrônicos.

Em 1987, a empresa expandiu seu escopo de atuação na área de desenvolvimento de sistemas de informação, focando, também, a indústria, recursos humanos (folha de pagamento), contabilidade e hospitais. Nesta mesma época a Hitachi, também, se propôs a desenvolver um software para gerenciamento de redes e aplicações (domínios de conhecimento classificados como ferramentas).

Atualmente, a empresa em questão desenvolve produtos que vão desde semicondutores, baterias, micro e mini-computadores a vários softwares nos mais variados domínios do conhecimento.

B.1.3 Caracterização da Toshiba

A Toshiba surgiu da união de duas empresas, ocorrida em 1939; uma de manufatura de telégrafos, fundada em 1875; e outra de produção de lâmpadas incandescente, fundada em 1890.

Em 1950, a Toshiba produzia computadores e semicondutores para todo o mercado Japonês.

Durante as décadas de 1960 e 1970 a empresa disputava uma competição nos mercados local e global, com uma ampla variedade de produtos manufaturados relacionados ao segmento de hardware e software. Este fato levou a empresa, na década de 1980, a se consolidar no mercado com a produção de minicomputadores, incluindo os *notebooks* (CUSUMANO (1991)).

A iniciativa de configurar uma fábrica de software na Toshiba partiu da divisão de produção de sistemas de controle industrial, cuja responsabilidade era produzir aplicações voltadas para indústria com alto grau confiabilidade.

Após consolidação da fábrica de software, a Toshiba entregava, na década de 1980, cerca de 50% de código reutilizável em seus softwares, isto levou a empresa a altos níveis de produtividade e qualidade.

Em 1987, a importância do software para empresa atingiu o seu pico, nesta época, a Toshiba contava com cerca de 11.000 funcionários no setor de desenvolvimento de software, sendo que 1.000 deles exerciam o cargo de engenheiro de software.

B.1.4 Caracterização da NEC

A NEC surgiu, em 1899, como uma subsidiária da ATT&T – *Western Electric*, se tornando independente logo após a sua criação.

Em 1958, a empresa iniciou o desenvolvimento e a comercialização de computadores transistorizados.

Por volta do ano de 1974, a NEC iniciou o desenvolvimento de sistemas operacionais para os computadores da família ACOS. Diferentemente, da Hitachi e da Toshiba, a NEC não se limitou a produzir software e hardware compatíveis com a tecnologia proposta pela IBM, fato este que caracterizou a empresa como uma alternativa de mercado por parte dos clientes.

No final da 1980, a NEC possuía em sua estrutura organizacional, cerca de 18.000 colaboradores.

Em 1987, a empresa se dividiu em dois seguimentos para atender os seguintes mercados:

- Processamento de informações: que inclui o desenvolvimento de sistemas de software caracterizados como aplicativos, e sistemas de informações automatizados com foco na área da indústria a gestão de operações;
- Produtos de comunicação: Desenvolvimento de produtos relacionados à transmissão de informações remotas.

Tal segmentação levou a empresa em questão a operar nos domínios relacionados a engenharia de produção, terminais de transmissão, rádio de comunicação e dispositivos eletrônicos.

Salienta-se que o seguimento de processamento de informação, estabelecido pela empresa, aglutina as subsidiárias caracterizadas como: especializadas e regionais. As especializadas são responsáveis pelo desenvolvimento de sistemas em diversas áreas do conhecimento, entre elas destacam-se: comunicação, aeroespaciais, oceânicos, informações gerenciais e telecomunicações. Já as subsidiárias regionais, caracterizadas como fábricas de software, se tornaram a base da produção distribuída da empresa, e o principal objetivo destas, é apoiar as subsidiárias especializadas na produção do software dentro de um contexto sistêmico.

Atualmente, a NEC possui uma ampla carteira de produtos e soluções focadas em consultoria de negócios. Dentro desta ótica é possível destacar os setores de comunicação de dados e voz; mobilidade; multimídia; computação de alto desempenho; computação baseada em servidor e segurança. Na área de sistema de informação automatizado a empresa provê soluções para os domínios relacionados à educação, finanças (bancos), hotelaria e saúde.

B.1.5 Caracterização da Fujitsu

A *Tsushinki Manufacturing Corporation*, companhia que se tornou a Fujitsu, nasceu da separação da *Fuji Electric* (empresa que fabrica máquina eletrônica) da *Siemens*.

Em 1950, a empresa produziu cerca de 5.000 telefones e, em 1954, a empresa começou a produzir computadores.

Por volta de 1962, a empresa se consolidou no mercado de hardware com a produção do FACOM 222, um computador transistorizado de propósito geral. Este produto começou a ser exportado para alguns países do continente asiático e a fama da empresa começou a chegar ao ocidente. Este último fato, levou a empresa a abrir uma subsidiária em *New York*, em 1967. A partir deste ano a empresa iniciou

um projeto de crescimento e, em 1988, já possuía cerca de 58.000 funcionários espalhados por vários países do mundo.

Devido ao grande crescimento e à entrada no mercado computacional, a empresa, em 1970, iniciou a padronização de seu processo de produção de software, culminando, em 1979, com a fundação de sua fábrica. Até 1988, a empresa desenvolvia softwares voltados aos seguintes segmentos de mercado: bancário, hospitalar, financeiro, governo e segurança.

Atualmente, a empresa provê soluções, relacionadas a software, nos mais variados domínios, entre eles é possível destacar: as aplicações para o gerenciamento de redes e banco de dados; o pacote integrado para a produção de software; ERP; CRM e comércio eletrônico.

Apresentada a caracterização das empresas que configuram este anexo, a próxima seção contempla as visões relacionadas aos “insumos do processo de produção” e “as atividades que o compõe o processo”.

B.2 O Processo de Produção de Software das Empresas

Esta seção aglutina duas visões do roteiro utilizado para o desenvolvimento do estudo de caso: a visão sob a ótica dos insumos e a visão das atividades do processo de produção.

Cabe ressaltar que o autor que embasa este anexo não apresenta, formalmente, em seu trabalho, questões sobre: o modelo do processo utilizado pelas empresas; os envolvidos com as atividades das visões gerencial e processual e; a caracterização do processo em relação a sua complexidade (leve ou pesado).

B.2.1 Processo de Produção da SDC

O processo da fábrica de software da SDC foi descrito, em 1976, reunindo as boas práticas existentes na empresa e as idéias advindas da área de engenharia de software configuradas na época. O processo em questão foca, principalmente, a estruturação das atividades de projeto de software e de codificação. A produção de uma biblioteca de programas também é uma constante abordada pelo processo.

O processo da fábrica de SDC aborda as seguintes práticas:

- promover a “*repetibilidade*” no processo de desenvolvimento de software;
- permitir a transferência rápida de conhecimento de um projeto para outro;
- estabelecer uma estrutura consistente para estimar custo;
- prover uma forte interação dos envolvidos com o processo, por meio de uma linguagem comum;
- prover a capacidade de medir o progresso do projeto;
- estabelecer um mecanismo para medir e assegurar a qualidade do software.

Nos relatos apresentados por CUSUMANO (1991), é perceptível que na empresa em questão, o processo foi dividido nas seguintes atividades: planejamento, levantamento de requisitos, análise de sistemas e projeto de software, desenvolvimento (codificação), teste e entrega. Na especificação das atividades, foram delineados os objetivos, as entradas, a função de transformação, as saídas e os critérios para avaliação de sucesso.

Desenvolvendo uma análise mais aprofundada nos relatos do autor citado, anteriormente, é possível perceber que a atividade de planejamento tem como

objetivos definir as estratégias para o desenvolvimento de software e o plano de métricas. Nela, as tarefas e os recursos necessários para o desenvolvimento do projeto são estabelecidos. O plano de projeto é caracterizado como principal artefato gerado por tal atividade, e sua finalidade é aglutinar informações relacionadas ao: plano de desenvolvimento de software; provimento de recursos humanos para execução do projeto; orçamento do projeto; técnicas utilizadas no gerenciamento de configuração; forma de organização da documentação gerada com a execução do processo; plano de garantia da qualidade e; monitoramento e controle do projeto.

Na atividade de levantamento de requisitos, os envolvidos com o processo devem delinear e descrever as características funcionais e não funcionais² do software. Alguns elementos inerentes ao ambiente de desenvolvimento de software são configurados nesta atividade, entre eles é possível destacar:

- Linguagem de programação na qual o software será desenvolvido;
- Padrões para o desenvolvimento de projeto de software. A linguagem para descrição do projeto e a granularidade das especificações das funcionalidades são definidas;
- Ferramentas que possam auxiliar o desenvolvimento do software (ou a execução das atividades do processo).

Percorrida a atividade de levantamento de requisitos, o software é projetado seguindo os elementos delineados. A atividade em questão objetiva:

- Apresentar de forma detalhada os requisitos funcionais do software. Salienta-se que neste detalhamento deve ser utilizada a linguagem para a descrição de projeto, lembrando que tal linguagem é estabelecida na atividade de levantamento de requisitos;

² Tempo de resposta, portabilidade de plataforma, usabilidade são caracterizados como requisitos não funcionais.

- Apresentar uma decomposição funcional do software (módulos e submódulos).

Ressalta-se que a atividade de projeto apresenta como resultado os componentes do software delineados sob a ótica dos modelos: de dados e funcional; da dependência e relacionamentos entre os módulos funcionais; da especificação de performance e; por fim, dos pontos de transições que serão utilizados para delinear a liberação do produto junto ao cliente.

Já a atividade de desenvolvimento tem como objetivo codificar as especificações, advindas da atividade de projeto. O reuso de código é uma constante no caso da SDC. Nesta atividade são identificadas as partes do produto que podem ser caracterizadas como reutilizáveis, e estas são desenvolvidas e disponibilizadas em uma biblioteca de funções (este fato traduz a idéia de “componentização”).

A atividade de teste tem como objetivo verificar se os módulos desenvolvidos estão de acordo com o que foi estabelecido na atividade de requisitos. Tal atividade inicia-se após o término da atividade de codificação e termina após a realização do teste de aceitação junto ao cliente.

Em fim, a atividade de entrega provê os subsídios necessários para a instalação do software, treinamento de pessoal e correção de possíveis erros encontrados. A adição de melhorias, também, é uma constante nesta atividade.

Salienta-se que modelo de organização da produção da SDC não possui características centralizadas, sua tônica foi configurada no desenvolvimento distribuído. Fato este justificável devido ao esquema de contratação do Departamento de Defesa Americano e de outros clientes. Tal esquema direcionava como requisito essencial, a contratação uma equipe de engenheiros, para trabalhar, internamente, objetivando caracterizar e definir as questões do projeto de software.

A produção distribuída acarretou alguns problemas para SDC, entre eles destaca-se: o entendimento das especificações, advindas da atividade de projeto, por parte dos programadores, o que leva a crer que a empresa em questão não possuía uma atividade de equalização bem definida e; a dificuldade em centralizar as informações para estabelecer um controle global dos projetos, devido à existência de várias equipes projetando software, junto aos clientes, e repassando os artefatos gerados para a codificação. Ressalta-se que as células de desenvolvimento encontravam-se descentralizadas. A Figura B.1 apresenta, visualmente, o modelo de organização de produção de software praticado pela SDC.

Por fim, a empresa em questão, é caracterizada sob a ótica do ciclo curto nas unidades produtoras de código, denominadas como equipe de programadores na Figura B.1. Já o ciclo longo se verifica sobre a perspectiva da equipe de projeto de software.

Figura B.1 - Modelo de organização da produção e delimitação dos ciclos da SDC (Figura construída com base nos relatos de CUSUMANO (1991))

B.2.2 Processo de Produção da Hitachi

O ciclo de produção da fábrica de software da Hitachi é caracterizado como longo (no o desenvolvimento de software para automatizar o sistema de informação

de uma determinada organização) e médio (se visto sob a ótica do domínio de conhecimento relacionado a ferramentas³).

As atividades de modelagem de negócio, projeto de software, codificação, teste (modular e integrado) e entrega do software (que inclui o teste de aceitação e os procedimentos relacionados à manutenção), fazem parte do processo de produção da empresa em questão.

Com o objetivo de modelar o negócio de uma determinada organização, a Hitachi tem como procedimento manter um grupo de pessoas trabalhando, diretamente, com seus clientes.

Um fato interessante no caso da Hitachi é a divisão dos analistas de sistemas por domínio do conhecimento, por exemplo: existe uma equipe responsável pelo desenvolvimento do modelo de negócio para a área da indústria e outra equipe responsável por desenvolver modelos para domínio bancário. Nos relatos de CUSUMANO (1991), também é possível destacar a questão da contratação interna, na qual uma determinada equipe responsável pela modelagem de negócios (ou análise de sistemas) contrata outra equipe para prestar os serviços de projeto de software, codificação e teste.

Na Hitachi as atividades do processo são divididas em tarefas. Para realização da modelagem de negócio é necessário compreender os eventos inerentes à estrutura sistêmica do cliente (tarefa 1) e materializá-lo utilizando uma linguagem de modelagem qualquer (por exemplo: um diagrama de fluxo de dados ou um fluxograma) (tarefa 2).

A atividade de projeto de software possui como tarefas: 1 – caracterizar o produto; 2 – determinar as especificações externas⁴ e internas⁵. Na atividade de

³ Neste caso uma ferramenta pode ser encarada como um editor de texto, um compilador ou uma planilha eletrônica. Produtos que não necessitam de uma análise sistêmica para o seu desenvolvimento.

⁴ Na determinação das especificações externas, a estrutura lógica do programa é elaborada, as camadas funcionais e as suas conexões são definidas.

codificação é possível verificar a tarefa da manufatura de código (compor os programas a partir das especificações utilizando uma biblioteca de funções pré-definida).

Já a atividade de teste congrega as tarefas para o desenvolvimento do teste modular (tarefa 1) e integrado (tarefa 2).

Por fim, a atividade de entrega aglutina as tarefas de demonstração do software para o cliente (tarefa 1), a implantação (tarefa 2) e o treinamento (tarefa 3).

Uma das preocupações da Hitachi é a questão da padronização dos artefatos relacionados ao processo. Fato este comprovado com a criação, em 1969, do comitê de padronização de software da Hitachi (*Hitachi Software Standard*). Tal comitê evoluiu até 1979 e configurou:

- um conjunto de padrões que compõem o processo de software;
- sistema de controle de componentes (biblioteca de programas);
- manual padrão para codificação;
- *checklists* para projeto e estimativas;
- modelo de registro de materiais técnicos e, por fim;
- um estudo de medidas de eficiência em engenharia de software.

Com base na descrição efetuada nesta seção, é possível inferir uma ilustração (vide Figura B.2) do processo de produção de software da empresa em questão.

⁵ Na determinação das especificações interna, a estrutura física do programa é definida, o arranjo hierárquico dos módulos, bem como, suas interconexões são estruturados. O modelo de dados também é desenvolvido.

Figura B.2 - Processo de produção de software da Hitachi (inferido a partir das considerações efetuadas por CUSUMANO (1991)).

B.2.3 Processo de Produção da Toshiba

O ciclo de produção da fábrica de software da Toshiba é caracterizado como longo, e possui as seguintes atividades:

- Levantamento de requisitos: cujo objetivo é definir, junto ao cliente do software (usuário), quais os eventos sistêmicos que irão compor o software e as características gerais do produto;
- Modelagem de negócio: responsável pela análise, otimização e documentação dos processos de negócio de uma determinada organização;
- Projeto de software: cujo objetivo é desenvolver e documentar todo o projeto de software, a partir da modelagem de negócio. Nesta atividade é possível visualizar, via documento de especificação, as funcionalidades e o relacionamento dos dados que compõem o software;
- Codificação: responsável pela produção do software a partir de documentos gerado na atividade de projeto. Nesta atividade, a Toshiba possui um forte conceito de reuso de código;

- Teste de software: responsável por verificar se os produtos desenvolvidos pela atividade de codificação possuem integridade funcional, configurada na atividade de projeto de software. A configuração da integridade funcional está baseada na especificação de requisitos, previamente, definida;
- Instalação ou Alinhamento: cujo objetivo é configurar o software nas estações de trabalho do cliente e desenvolver um treinamento (conceito definido como alinhamento dentro da Toshiba) para utilização do produto;
- Manutenção: tal atividade é disparada, somente, quando um determinado problema no software é detectado. São considerados problemas: deficiências técnicas relacionadas ao funcionamento e a alteração de requisitos funcionais (alterações estas advindas do ambiente, por exemplo: alteração de uma determinada lei).

A apresentação gráfica do relacionamento entre as atividades do processo de software da Toshiba pode ser visualizada por meio da Figura B.3.

Em relação ao estabelecimento das equipes para a execução das atividades pertinentes ao processo, a Toshiba, pratica a orientação a domínio de conhecimento na visão de negócio do projeto, por exemplo: Caso um analista esteja modelando o sistema de uma indústria automobilística, o mesmo deve ter conhecimentos aprofundados sobre os aspectos organizacionais para tal. Já os programadores, pessoas responsáveis pelas atividades de codificação e teste, não necessitam possuir especialização em relação ao domínio do conhecimento, o que possibilita a flexibilização e um reaproveitamento de pessoal conforme a demanda de projetos.

Figura B.3 - Processo de produção de software da Toshiba (inferido a partir das considerações efetuadas por CUSUMANO (1991)).

Um outro fato destacado por CUSUMANO (1991), no estudo de caso da Toshiba, é a presença de uma linguagem formal para a descrição dos requisitos. Com os requisitos documentados em tal linguagem é possível submetê-los a um processo transformacional gerando, automaticamente, o documento de projeto de software (documento este também expresso em uma linguagem formal e apoiado pelos diagramas advindos do SADT⁶). No documento de projeto de software é possível encontrar as funcionalidades do produto, e a partir dele são gerados os programas (algo parecido com as ferramentas case).

B.2.4 Processo de Produção da NEC

O processo de produção de software da NEC tem como objetivo prover um alto grau de padronização e automatização das atividades e tarefas, primando sempre pela qualidade, tanto sob a ótica do processo quanto do produto. Dentro desta linha, a empresa em questão dividiu o seu processo de produção em 6 atividades: planejamento, projeto básico, projeto detalhado, implementação, integração, inspeção e manutenção (atividade, também, responsável pela entrega do software), todas elas detalhadas na Tabela B.1.

⁶ Structured Analysis Design Technique (definição já apresentada no capítulo 5)

Atividade	Objetivo	Entrada	Saída (documentos)
Planejamento (gerenciamento de projetos)	Documentar o produto baseado nas necessidades do mercado.	Características do produto. Necessidades do mercado.	Plano de projeto. Cronograma. Planilha de Custo
Projeto Básico L(projeto de software)	Analisar os objetivos do plano de projeto e desenvolver o plano para produção do software caracterizando, assim, o produto.	Plano de projeto. Cronograma. Planilha de custo.	Plano de produção do software. Requisitos de desempenho. Revisão do cronograma e do custo.
Projeto detalhado. (projeto de software)	Detalhar as funções (ou módulos) baseado no plano de produção do software.	Plano de produção do software.	Especificação lógica das funções. Revisão do cronograma e do custo.
Implementação (produção do código – teste unitário)	Codificar e testar cada função (ou módulo) detalhada na especificação lógica.	Especificação lógica das funções.	Manual do usuário, código implementado e testado. Revisão do cronograma e do custo e das especificações das funções.
Integração (teste integrado e entrega)	Integrar as funções e módulos. Testar a integração.	Código implementado e testado.	Documento de teste relacionados à integração das funções ou módulos. Revisão de desempenho do produto. Software pronto para ser entregue.
Inspeção (gerenciamento de projeto – controle)	Verificar se o plano de projeto foi executado sob a ótica do cronograma e do custo.	Plano de projeto. Cronograma. Planilha de custo. Tempo e recursos para execução das atividades.	Documento de inspeção para armazenamento do histórico de projeto.
Manutenção	Melhorar e prover suporte ao produto no mercado.	Solicitações de melhorias	Melhorias encaradas como um novo projeto.

Tabela B.1– Especificação das atividades pertinentes ao processo de produção de software da NEC. Inferido a partir das considerações efetuadas por CUSUMANO (1991).

Ao analisar as atividades do processo de produção de software da fábrica, é possível afirmar que o ciclo de produção implementado pela NEC é classificado como médio. Tal relação pode ser verificada na primeira coluna da Tabela B.1, no texto grafado logo após o nome da atividade do processo. As atividades pertinentes ao ciclo longo do processo de produção de software não são praticadas por tal empresa, devido a dois fatos: primeiro, a empresa foca a maior parte de seu domínio de conhecimento em produtos caracterizados como ferramentas e não em sistemas de informação automatizados; segundo, a área que provê soluções relacionadas aos sistemas de informações automatizados é bastante restrita, atendendo, somente, as necessidades das empresas de educação, finanças (bancos), hotelaria e saúde.

Por fim, a NEC utiliza, desde 1970, uma estrutura distribuída de produção de software, sendo que, as regras e as ferramentas pertinentes ao processo são definidas por um laboratório, denominado como Laboratório Central para Estabelecimento do Processo de Software. É importante ressaltar que as unidades satélites, caracterizadas como fábricas de software, compartilham, remotamente, o processo, as informações sobre projetos concluídos e as ferramentas utilizadas para o provimento de soluções para clientes nos segmentos já elucidados.

B.2.5 Processo de Produção da Fujitsu

Em 1979, a Fujitsu iniciou as atividades de sua fábrica de software para o desenvolvimento de sistemas automatizados voltados aos seguintes domínios do conhecimento: bancos, empresas de seguranças, governo e manufatura.

O processo fabril para o desenvolvimento da empresa em questão, foi dividido nas seguintes atividades:

- Planejamento do projeto: Nesta atividade são mapeados o custo, os recursos e o cronograma necessário para o desenvolvimento do software;
- Análise de sistemas: Atividade responsável pela materialização do modelo sistêmico, na qual o software irá aderir;
- Projeto da estrutura do software: Atividade cujo objetivo é apresentar as características globais do software, incluindo as funcionalidades;
- Projeto modular: Nesta atividade, as funcionalidades são divididas em componentes e estes são especificados;
- Programação: Atividade responsável pela codificação dos componentes definidos pela atividade de projeto modular;

- Teste modular: Atividade que utiliza o plano de teste, definido na atividade de projeto modular, para verificar se os componentes produzidos estão corretos e podem ser integrados;
- Teste de Integração: Atividade que utiliza o plano de teste, definido na atividade de projeto da estrutura do software, para verificar a existências de erros no momento da integração dos componentes;
- Teste de sistema: Atividade cuja responsabilidade é verificar se software adere aos eventos sistêmicos, selecionados para automatização;
- Teste operacional: Classificado como teste de aceitação, esta atividade conta com a participação do grupo de usuários do software, e é realizada no local e no equipamento que o software irá executar;
- Entrega: Instalação do software, junto ao cliente. Em tal atividade, também, são contempladas questões de treinamento;
- Manutenção: Melhorias e manutenções corretivas são desenvolvidas nesta atividade.

A relação das atividades do processo de produção de software da fábrica da Fujitsu pode ser verificada por meio da Figura B.4.

Figura B.4 – Relação das atividades do processo de produção de software – Fujitsu (Adaptado de CUSUMANO (1991) página 358)

Por fim, ao analisar a figura acima é perceptível que o processo de produção da fábrica de software em questão evoluiu em relação aos ciclos de produção. Em 1979, a empresa organizou as atividades pertinentes ao ciclo curto. Já, em 1982, as atividades, pertinentes ao ciclo médio, foram definidas e incorporadas ao processo e, por fim, em 1985, a empresa adicionou as atividades relativas ao ciclo longo do processo.

B.3 Gestão Operacional das Fábricas

Nesta seção serão apresentados os aspectos relacionados à gestão operacional das fábricas de software, nominalmente, citadas no início deste anexo.

B.3.1 Gestão Operacional da SDC

A gestão operacional da fábrica de software da SDC contempla questões relacionadas ao gerenciamento de projetos e da qualidade.

Em relação ao gerenciamento de projetos, foi verificado na seção B.2.1, que a empresa em questão desenvolve a atividade de planejamento, cujo objetivo é definir questões relacionadas a: recursos humanos e financeiros para execução do projeto dentro do cronograma estabelecido. Para a realização desta atividade, a empresa conta com um escritório de projetos. Tal escritório, de posse das informações históricas dos projetos desenvolvidos, estabelece as métricas relacionadas aos recursos e o tempo necessário para o desenvolvimento do produto. Além da atividade de planejamento, a SDC pratica, constantemente, tarefas relacionadas ao controle de projeto cujo objetivo é verificar se os recursos estabelecidos para o projeto são suficientes.

Já a gestão da qualidade é feita sob a ótica do processo e do produto. Na primeira, as informações geradas pelas atividades do processo devem estar consistentes e os artefatos devem estar preenchidos de forma correta. A gestão da qualidade do produto é feita, através da satisfação do cliente para com o mesmo.

B.3.2 Gestão Operacional da Hitachi

A atividade de gestão de projetos da Hitachi pode ser dividida em três grandes tarefas: planejamento; execução e controle.

Na tarefa de planejamento artefatos como cronograma, planilha de custos e planilha de recursos humanos, para o desenvolvimento do software, são mapeados. A tarefa de execução objetiva que as atividades do processo sejam desenvolvidas. A atividade de controle verifica se o que foi planejado está sendo executado. Salienta-se, também, que nesta tarefa os artefatos, inerentes a tarefa planejamento, podem ser alterados.

A métrica utilizada na atividade de controle de projeto está baseada no acoplamento de programas a um determinado módulo e no acoplamento de módulos a um determinado projeto, por exemplo: sabendo que um projeto P foi particionado em 2 módulos (M_1 e M_2) e sabendo, também, que o módulo (M_1) possui 100 programas, sendo que 77 destes estão finalizados, a gerência de projeto conclui que cerca de 80% do módulo está finalizado.

Além do gerenciamento do projeto, existe a atividade de gerência da qualidade. Este tipo de gestão tem como objetivo controlar e garantir que o produto de software produzido irá atender as expectativas dos clientes, minimizando assim a quantidade defeitos.

A métrica utilizada na gerência da qualidade totaliza o número de erros (diário e mensal) gerado por programador, por tipo de programa e pela linguagem utilizada. Esta métrica é utilizada na prevenção de defeitos, pois a mesma pode espelhar que um determinado tipo de programa pode estar mais sujeito a erros.

Na Hitachi, a política de análise de confiabilidade estabelecida prega que:

- para cada conjunto de programas é necessário informar o número de erros esperados e confrontar com os números de erros encontrados;
- utilizar um *ckecklist* para assegurar que a execução dos componentes de código atendam os requisitos apresentados na especificação de projeto e, por fim;
- analisar o erro encontrado confrontando questões como tamanho do programa, estrutura e número de horas gastas no trabalho.

As informações mapeadas na análise de confiabilidade são armazenadas e, posteriormente, utilizadas em outros projetos com o objetivo de minimizar os defeitos aumentando, assim, a qualidade do produto software desenvolvido pela Hitachi.

Com a implantação dos processos de gestão (de projeto e da qualidade), a Hitachi, obteve um bom resultado em relação aos projetos atrasados e à média de defeitos (mensal) apontados pelos clientes caiu. Os números que comprovam esta afirmação estão expressos na Tabela B.2 e na Tabela B.3.

Ano	(%) Projetos Atrasados*	Ano	(%) Projetos Atrasados*
1970	72,4	1978	9,8
1971	56,6	1979	7,4
1972	36,3	1980	10,7
1973	13,0	1981	14,0
1974	6,9	1982	12,9
1975	9,8	1983	18,8
1976	16,8	1984	16,3
1977	16,1	1985	18,0

Tabela B.2 – Projetos de software atrasados.
*Baseada no cronograma original. (Extraída de CUSUMANO (1991) página 187).

Ano	*Média de Defeito mensal
1978	100
1979	79
1980	48
1981	30
1982	19
1983	13
1984	13
1985	14

Tabela B.3 – Média de falha de sistemas apontados pelos clientes (*estações por mês).
Extraída de (CUSUMANO (1991) página 191).

Por fim, as informações sobre os aspectos da gestão de configuração, da empresa em questão, não são aprofundadas por CUSUMANO (1991).

B.3.3 Gestão Operacional da Toshiba

Esta seção tem como objetivo apresentar a configuração da gestão de projetos da fábrica de software da Toshiba.

Estruturalmente, a Toshiba possui 4 áreas de trabalhos: centro de serviço da bancada de trabalho de software⁷; centro de serviço de documentação; sala de armazenamento de arquivos⁸; instalações para teste completos unindo dispositivos de hardware ao software. Cada uma destas áreas, excluindo a sala de armazenamento de arquivos, possui um terminal, denominado como estação de trabalho (*workstation*), conectado ao banco de dados central (neste banco estão

⁷ A bancada de trabalho de software tem como objetivo suportar as ferramentas de projeto, codificação e teste no desenvolvimento do software (CUSUMANO (1991)).

⁸ A sala de armazenamento de arquivos tem como objetivo guardar, fisicamente, os arquivos gerados pelo centro de serviço de documentação. Atualmente, esta sala poderia ser substituída pelo uma biblioteca digital.

armazenadas informações sobre projeto de software, incluindo métricas). Esta estrutura pode ser replicada para cada projeto a ser desenvolvido pela Toshiba.

Um determinado projeto pode envolver várias estações de trabalhos, sendo que a maioria dos projetos pertencentes à Toshiba possuem de 4 a 5 analistas de sistemas, na qual estes são responsáveis pela modelagem de negócio; de 10 a 15 engenheiros trabalhando no projeto de software; e de 70 a 80 programadores trabalhando nas atividades de codificação e teste.

Os responsáveis pela atividade de gestão de projeto, dividem o projeto de software em atividades (respeitando descrição apresentada na seção B.2.3) e as unidades de trabalhos são responsáveis pela execução de um conjunto de tarefas.

É Importante salientar que o planejamento de projeto tem como objetivo dividir o projeto em atividades, e configurar as unidades de trabalho. Todo o conceito de planejamento da empresa é suportado por uma base histórica de projetos, na qual são armazenadas informações dos projetos em andamento e dos projetos finalizados. Quando novos projetos são concebidos, tal base é consultada, com o objetivo de minimizar distorções numéricas em relação à concepção de custo e prazo.

Um outro fator importante na gestão do projeto, praticada pela Toshiba, é a questão do monitoramento em relação à produtividade. Tal questão é ajustada aos níveis de experiência dos envolvidos com o projeto. Envolvidos mais experiente, com certeza devem produzir mais. A produtividade dos envolvidos com o projeto pode ser configurada por dia, por semana ou por mês, a forma desta configuração, fica sob a responsabilidade do gerente de projeto.

O controle de produtividade utilizado pela fábrica de software da Toshiba é feito através das seguintes métricas: total de linhas de código entregue por pessoa em determinado período e o percentual de reuso do software. A empresa utiliza o

conceito de Linhas de Código Equivalentes em *Assembly*⁹ (LCEA) para ajustar a sua métrica. A Tabela B.4, apresenta a produtividade anual (de 1972 a 1985) da fábrica de software em questão.

Tal tabela pode ser analisada sobre duas óticas: antes da criação da fábrica de software (de 1972 a 1976) e após a criação da fábrica de software (1977 a 1985).

Ano	Total de LCEA entregue (pessoa / mês)	Crescimento do índice de produtividade (%) [*]	Varição do índice de produtividade (%) ^{**}	% de reuso ^{***}	Novo código (LCEA) ^{****}	Defeitos por 100 LCEA	Número de funcionários
Dados antes da criação da fábrica de software							
1972	1230						
1973	1390	13	+13				
1974	1370	11	-2				
1975	1210	-2	-12				
1976	1390	13	+15				
Dados depois da criação da fábrica de software							
1977	Dado não disponível						
1978	1684	37				7-20	1200
1979	1988	62	+18	13	1730		1500
1980	2072	68	+4	16	1740		1700
1981	2443	99	+18	29	1735		2050
1982	2595	110	+6	26	1920		2100
1983	2763	125	+7	41	1630		2150
1984	2931	138	+6	45	1612		2250
1985	3130	154	+7	48	1612	0,2-0,05	2300

Tabela B.4- Produtividade da fábrica de software da Toshiba (Retirada de Cusumano (1991) página 240). *Crescimento do índice de produtividade em relação a 1972. **Calculo da variação efetuada sobre o ano anterior. ***Reuso: Entrega de software com até 20% de modificação. **** Calculo obtido através de LCEA * [(100 - %reuso)/100]

Em 1972, a média de LCEA entregue por pessoa era de 1230, em 1973 houve um aumento de 13%, decaindo em 2%, em 1974, e 12%, em 1975, e, por fim, estabilizando-se, novamente, em 1976. Com a criação da fábrica de software, em 1977, a Toshiba, logo no seu primeiro ano obteve um crescimento de 37% na média de LCEA entregues por pessoa, culminando com 154% em 1985. O percentual de reuso praticado pela Toshiba, neste mesmo ano, culminou a 48%. Os altos níveis em relação ao reuso de software elevaram a qualidade do produto desenvolvido na Toshiba, pois quando o código torna-se reutilizável indica, que o mesmo está quase, que totalmente, estabilizado.

⁹ A Toshiba possui uma tabela que converte instruções de alto nível em linhas de código em *Assembly*.

A atividade que tem uma grande influência na qualidade do software da Toshiba, principalmente, na questão do produto, é o teste. Nela o software é testado de três formas: unitário (somente um componente ou uma função é testado); teste de integração (componentes acoplados são testados) e; por fim o teste de requisitos (o software é testado de acordo com os requisitos coletados). Quando o teste de requisitos é efetuado é possível verificar que a Toshiba desenvolve o conceito de rastreabilidade¹⁰ (IEEE 830-1998) (vide Figura B.5).

Figura B.5 - Rastreabilidade no processo de teste para controle da qualidade (Toshiba) – Inferido a partir das considerações efetuadas por CUSUMANO (1991).

Por meio da Figura B.5 é possível verificar que um determinado requisito A, pode se dividir em n processos de negócio (ou processos sistêmicos), um processo sistêmico pode se dividir em m processos de software e um processo de software pode se dividir em i componentes ou partes do código. Tais componentes são implementados e testados em relação à sua funcionalidade. No teste integrado, a unidade software (parte do software desenvolvido) deve estar de acordo com os requisitos estabelecidos pelo cliente da fábrica. Para garantir que a unidade de software atenda um determinado requisito é necessário garantir que os artefatos

¹⁰ O conceito de rastreabilidade é denominado de gestão de teste e qualidade pela Toshiba.

relacionados àquele requisito, gerados nas atividades anteriores, estejam corretos e consistentes.

Por fim, o conceito de rastreabilidade, implementado na Toshiba, provê a fábrica à tentativa de capturar um determinado erro antes de atingir a atividade de teste. Tentativa esta desenvolvida por meio da simulação de cenários (ou revisões de artefatos), praticadas nas várias atividades de projeto de software. Tal simulação tem como objetivo verificar o ambiente no qual o software irá operar e verificar se os requisitos, apresentados pelo cliente da fábrica de software, estão sendo respeitados.

B.3.4 Gestão Operacional da NEC

As atividades ligadas à gestão operacional da NEC foram configuradas, em 1976, com a criação de uma divisão central, cujo objetivo era aglutinar as informações sobre qualidade e produtividade de todas as subsidiárias, que trabalhavam de forma distribuída em relação produção de software. Para atingir tal objetivo, a NEC teve que trabalhar, fortemente, na:

- padronização de ferramentas, principalmente, aquelas ligadas à gestão de projetos e da qualidade.
- padronização das atividades de desenvolvimento de software, atividades estas descritas, anteriormente.
- padronização de componentes de código, para que estes pudessem ser, amplamente, utilizados no processo de produção.
- reunião das informações em um banco de dados, visto que uma das características da empresa em questão é o desenvolvimento de software de forma distribuída.

Cabe ressaltar que após o processo de padronização, a empresa em questão criou o *Standardized Technology and Engineering for Programming Suport* (STEPS), documento caracterizado como um guia aglutinador, que contém a descrição formal, de todas as atividades, tarefas e ferramentas a serem utilizadas para a produção de software¹¹.

Por meio do processo de padronização e com a criação do documento de processo a NEC passou a planejar, executar e controlar seus projetos com maior afinho e destreza. Ressalta-se que a tarefa de controle¹² está, totalmente, baseada na execução das atividades do processo de produção, ficando sob responsabilidade do planejamento estabelecer as estimativas necessárias para o desenvolvimento do projeto. Os dados estimados e gerados para planejamento e controle do projeto, podem ser verificados por meio da Tabela B.5.

Com a atividade de planejamento intensificada, a NEC começou a colher o resultado propiciado pelos esforços praticados a partir de 1976. Entre eles é possível destacar:

- a melhoria da produtividade;
- o aumento maciço do conceito de reuso de software, e;
- a melhoria no processo de estimativas (custo, prazo e recursos) para o desenvolvimento de novos projetos.

¹¹ Atualmente, o documento que descreve, formalmente, o processo de produção de software é denominado pelas empresas como “documento de processo”.

¹² Com base nos dados colhidos, a tarefa de controle, tem como meta ajustar o plano de projeto.

Atividade: Planejamento de Projeto – Estabelecimento de estimativas. Estimativa de cronograma (dividido em atividades e tarefas). Estimativa de tamanho do programa. Estimativa de custo. Experiência necessária, por parte do programador, para desenvolver o projeto. Nível de dificuldade que o processo poderia exigir	P L A N E J.
Atividade: Projeto básico – Dados utilizados no controle do projeto Data de início e de término da atividade. Diferença entre o orçado e realizado, cronograma e custo. Qualidade das especificações funcionais geradas (aspecto definido pelo comitê de controle da qualidade). Recursos (humanos) utilizados para o desenvolvimento da atividade..	C O N T R O L E
Atividade: Projeto detalhado – Dados utilizados no controle do projeto Data de início e de término da atividade. Diferença entre o orçado e realizado, cronograma e custo. Qualidade das especificações funcionais geradas (aspecto definido pelo comitê de controle da qualidade). Recursos (humanos) utilizados para o desenvolvimento da atividade.	
Atividade: Implementação – Dados utilizados no controle do projeto Data de início e de término da atividade. Diferença entre o orçado e realizado, cronograma e custo. Tamanho real do programa. Recursos (humanos) utilizados para o desenvolvimento da atividade. Inspeção do código (padronização de código fonte).	
Atividade: Teste Unitário – Dados utilizados no controle do projeto Número de casos de testes executados Data de início e de término da atividade. Diferença entre o orçado e realizado, cronograma e custo. Quantidade de erros encontrados. Tempo que cada caso de teste levou para ser executado. Recursos (humanos) utilizados para o desenvolvimento da atividade.	
Atividade: Teste Integrado – Dados utilizados no controle do projeto Número de casos de testes executados Data de início e de término da atividade. Diferença entre o orçado e realizado, cronograma e custo. Quantidade de erros encontrados. Tempo que cada caso de teste levou para se executado. Recursos (humanos) utilizados para o desenvolvimento da atividade.	

Tabela B.5 – Especificação das tarefas de planejamento e de controle da atividade de gestão de projetos da NEC. Adaptado de CUSUMANO (1991).

Além da organização da área de gestão de projetos, a NEC não mediu esforços para melhorar questões relacionadas à qualidade do produto e do processo. Dentro desta ótica a empresa criou, na segunda metade da década de 1970, o comitê de controle da qualidade, cujo objetivo era:

- verificar se o documento do processo estava sendo respeitado pelos colaboradores;
- verificar se os componentes produzidos respeitam os padrões de conectividade, estipulados pela empresa, e;

- coleccionar dados para análise estatísticas, entre estes destacam-se: quantidade de erros encontrados na atividade de testes; ordens de serviços corretivas, geradas após a entrega do produto.

Com a criação de tal comitê e com aplicação do processo de produção, a NEC obteve, na época, vários resultados na qualidade de seus produtos, alguns deles podem ser verificados por meio Tabela B.6. Ao analisar tal tabela, é possível verificar que a quantidade de defeito em produtos relacionados ao domínio de conhecimento “controle de transmissão” caiu de 1,37 para 0,47 klc com o desenvolvimento do programa de melhoria da qualidade.

Divisão (Subsidiária)	Objetivo de qualidade a ser analisado pelo comitê	Resultado	
		Antes da criação comitê	Após a criação do comitê
Controle de transmissão.	Defeito médio	1,37 / klc	0,47 / klc
Sistema de informação automatizado operando em minicomputadores.	Defeito médio	0,35 / klc	0,25 / klc
Grandes aplicações.	Alteração nas especificações	Abaixo de 40 %	

Tabela B.6 – Resultados aferidos após o desenvolvimento do programa de qualidade da NEC (Adaptado de CUSUMANO (1991)). Legenda: klc (1000 linha se código).

Um outro controle (representado na Tabela B.7) utilizado pela NEC é o relacionamento dos critérios de qualidade do software com os critérios de qualidade de requisitos. Nele é possível verificar questões relacionadas a rastreabilidade dos requisitos, consistência, simplicidade de especificação, modularidade, entre outras (IEEE 830-1998). Estas questões são relacionadas com critérios como “corretude” (C), confiabilidade (CO), manutenibilidade (M), flexibilidade (F), usabilidade (U), eficiência (E), segurança (S).

Questões (de qualidade de software) verificadas	Critérios da qualidade de requisitos						
	C	CO	M	F	U	E	S
Rastreabilidade	X						
Consistência	X	X					
Simplicidade		X					
Precisão		X					
Tolerância a Erros		X					
Modularidade			X	X			
Auto-descritivo			X	X			
Consistência			X				
Instrumentação			X				
Generalidade				X			
Expansível				X			
Treinamento					X		
Comunicação					X		
Operabilidade					X		
Independência da Máquina			X	X			
Independência de Software			X	X			
Eficiência de Execução						X	
Eficiência de Armazenamento						X	
Controle de Acesso							X
Compartilhamento de dados							X
Compartilhamento de comunicações							X

Tabela B.7 – Relação entre os critérios da qualidade de software e qualidade dos requisitos (Adaptada de CUSUMANO (1991)).

Por fim, questões relacionadas à gestão de configuração também são praticadas pela fábrica de software em questão, porém estas não são aprofundadas no texto de CUSUMANO (1991).

B.3.5 Gestão Operacional da Fujitsu

A gestão operacional da fábrica de software da Fujitsu se divide em duas atividades: planejamento (definida na seção B.2.5) e controle.

A atividade de controle tem como objetivo verificar se o cronograma e os recursos definidos na atividade de planejamento estão sendo respeitados durante a execução do processo de software. É importante salientar que se necessário, o gerente de projeto pode alterar o plano de projeto, ajustando-o em relação ao tempo, custos e recursos.

Além de um controle global, como explicitado no parágrafo anterior, a Fujitsu divide o projeto em fases, respeitando as atividades definidas no processo, e

gerencia com exclusividade. Com isto, pode-se concluir que os dados gerados, com a execução de cada atividade do processo, quando unidos, proporcionam ao gerente uma visão macro sobre os aspectos gerenciais.

Um outro aspecto, não menos importante, em relação à gestão de projeto de software praticada pela Fujitsu, está, intimamente, ligado a questão das medidas de produtividade. A produtividade da fábrica é norteadada pela quantidade de linhas de código produzidas por programador em um determinado período (por hora, por dia ou por mês); ou pelas horas utilizadas por engenheiros ou projetistas na execução das outras atividades do processo¹³. Um exemplo dos valores extraídos pela Fujitsu com a implantação das atividades relacionadas e a gestão operacional pode ser verificado por meio da Tabela B.8.

Métrica aferida	Antes da implantação do conceito de reuso de software	Após a implantação do conceito de reuso de software
Tamanho de todos os softwares (linhas de código)	3.600.000	8.300.000
Código reutilizado (linhas de código)	1.080.000 (30%)	6.550.000 (79%)
Código Novo (linhas de código)	2.520.000	1.750.000
Produtividade (Linhas de código/Mês de trabalho)	450	902

Tabela B.8 – Produtividade do software para o domínio bancário. (Extraído de CUSUMANO (1991) página 382).

Ao analisar tal tabela, é possível perceber que com a implantação da fábrica de software e com institucionalização das atividades relacionadas a gestão operacional, a Fujitsu passou a produzir mais linhas de código, utilizando a técnica de reuso.

Além das medidas relacionadas à produtividade, a fábrica de software da Fujitsu, também está preocupada com a questão da qualidade do processo e do produto. A primeira tem como principal foco verificar se as atividades e as tarefas do processo estão sendo executadas conforme a sua especificação. Já a segunda tem como meta verificar a qualidade do produto entregue para o cliente. A principal métrica da qualidade do produto, definida pela empresa em questão, é o número de

¹³ Por exemplo, um grupo de engenheiros de software levou cerca de 20 horas para desenvolver o projeto modular de uma determinada parte do software.

defeitos por linha de código. A área de qualidade da fábrica de software da Fujitsu consegue mapear números como os expressos na Tabela B.9.

Ao analisar tal tabela é possível perceber que a qualidade do produto, se verificada sob a ótica da quantidade de erros reportados pelos usuários, vem aumentando. Em 1977 a fábrica de software corrigia cerca de 0.19 defeitos em 1000 linhas de código programadas, sendo que a maioria delas, cerca de 85% eram detectadas na atividade de teste e nenhum era detectado a partir da revisão do projeto de software. Já em 1982, o índice de erros caiu para 0.01, sendo que 30% eram detectados por meio da atividade de teste, outros 30, por meio da revisão de código fonte (teste de caixa branca) e 40% detectados à partir da revisão de projeto do projeto de software. Este fato indica que partes dos erros detectados se concentram em problemas de entendimento das especificações do projeto.

Ano	Defeitos/1000 linhas de código mantidas.	Método de Detecção de erros (estimativas)		
		Teste (%)	Revisão de código fonte (%)	Revisão de folha de projeto (%)
1977	0.19	85	15	
1978	0.13	80	15	5
1979	0.06	70	20	10
1980	0.05	60	25	15
1981	0.04	40	30	30
1982	0.02	30	30	40
1983	0.02	--	--	--
1984	0.02	--	--	--
1985	0.01	--	--	--

Tabela B.9 – Defeitos em linhas de código reportados pelos envolvidos com o processo e pelos usuários – Extraído de CUSUMANO (1991) página 352.

Por fim, questões relacionadas à gestão de configuração também são detectadas, superficialmente, na fábrica de software em questão, porém estas não são aprofundadas na bibliografia utilizada para o desenvolvimento deste anexo.

B.4 As Ferramentas e Questões sobre o Armazenamento de Dados

As fábricas de software citadas, nominalmente, no início deste anexo possuem em sua estrutura um conjunto básico de ferramentas utilizadas no desenvolvimento e controle das atividades pertinentes às visões: gerencial e processual. De posse desta informação esta seção descreve as ferramentas utilizadas pelas empresas.

Seguindo a mesma sistemática, na descrição de tal visão, concebida no anexo A, é possível afirmar que as ferramentas utilizadas pela SDC podem ser inseridas em ambas visões organizacionais: gerencial e processual.

Na visão gerencial a empresa em questão utiliza uma ferramenta de gestão de projetos, cuja responsabilidade é capturar a produtividade dos envolvidos com o processo. Tal ferramenta não está integrada às ferramentas de produção, pertinentes a visão processual. Na alimentação de tal ferramenta, os dados sobre produtividade, são inseridos, manualmente, pelos envolvidos com o processo de produção. Tais dados são analisados, periodicamente, pelo gerente de projeto, armazenados em uma base histórica e utilizados em estimativas de recursos, de custos e de cronograma em novos projetos.

Já a visão processual, além das ferramentas que relacionam, diretamente, com o processo de produção, é permeada por uma ferramenta estruturada, cujo objetivo é armazenar componentes de código (funções, procedimentos e estruturas que possibilitam reutilização na construção de uma unidade de software ou funcionalidade). Esta ferramenta provê o acesso formal a tais componentes, mapeando informações relacionadas à sua forma de utilização, versão e responsável pelo desenvolvimento.

A Hitachi, por sua vez, possui uma ferramenta CASE (*Computer-Aided Software Engineering*) aderente às atividades de projeto de software, codificação e

teste. Tal ferramenta é dividida em três módulos, cuja descrição é apresentada pelos tópicos abaixo:

- Módulo de projeto: Tem como objetivo analisar, automaticamente, as especificações de software advindas do ambiente externo em relação à padronização dos artefatos;
- Módulo de programação: Tem como objetivo propiciar a compilação de um programa na linguagem HPL (adaptação da linguagem PL/1 desenvolvida pela própria Hitachi), respeitando a especificação modular do documento de projeto;
- Módulo de teste: Suporta o teste unitário e integrado. Gera, automaticamente, os casos de teste, com base em uma especificação externa, objetivando, assim, a configuração das unidades de teste que serão processadas na execução da atividade em questão.

Além do CASE, a Hitachi possui uma outra ferramenta denominada *Computer-Aided Software Production System* (CAPS), cujo objetivo é alimentar um banco de dados gerencial durante o desenvolvimento do projeto. Dados como custo do projeto, tempo de trabalho despendido para execução de uma tarefa e o progresso dos envolvidos com o projeto em relação ao mesmo são capturados, automaticamente, por esta ferramenta. Ao analisar o conjunto de dados armazenados pela ferramenta é possível concluir que a mesma tem como foco acompanhar a produtividade da fábrica de software da Hitachi (visão gerencial). Este fato levou a Hitachi integrar o CAPS junto ao CASE proporcionando, assim, a automatização na coleta das informações relacionadas à produtividade, pois ao instanciar um módulo do CASE, dados sobre o tempo de execução dos módulos (de projeto, de programação e de testes) são capturados pela CAPS.

Já o conjunto de ferramentas desenvolvidas pela Toshiba é definido pela própria empresa como bancada de trabalho em software. O objetivo da bancada é prover infra-estrutura e técnicas para análise de sistemas (modelagem de negócio),

projeto de software, programação, testes e gerenciamento de projetos. A bancada provê suporte a:

- Definição dos requisitos e o projeto básico do sistema;
- Construção de programas e detalhamento dos módulos de projeto (codificação);
- Controle de projeto em relação à produtividade, custo e cronograma;
- Controle da qualidade do produto: Controle este feito por meio dos dados gerados com a execução das atividades de teste (média de erros por programador, por exemplo);
- Manutenção de programas e reusabilidade.

As ferramentas compartilham informações por meio de um banco de dados, onde este também é utilizado para a composição das estimativas de projetos que estão em fase inicial de desenvolvimento.

A NEC, por sua vez, também, possui dentro de sua fábrica de software, seis ferramentas para a automação de seu processo produtivo (além da ferramenta de gestão de projeto). Entre elas é possível destacar:

- Ferramenta de prototipagem: Define, interfaces externas para entrada e saída de dados, por exemplo: telas, menus de acesso a informações e *layout* de relatórios;
- Ferramenta de projeto de software: gera diagramas e fluxogramas para a composição do documento de arquitetura do software;
- Ferramenta de manufatura: Gerador de código utilizado para automatizar a produção de código da fábrica de software em questão;

- Ferramenta de teste: Caracterizada como um gerador de casos e dados de testes. Utiliza como parâmetros às especificações advindas da atividade de projeto de software;
- Ferramenta de gestão de projeto: Provê subsídios para que a produtividade dos envolvidos com o projeto seja acompanhada. As informações geradas são armazenadas em uma base de dados, na qual esta serve como fonte para definição de estimativas;
- Ferramenta para controle de versões. Utilizada para controlar as versões dos documentos (ou artefatos) gerados com a execução do processo de software para um determinado projeto.

Por fim, o conjunto de ferramentas utilizado para automação das tarefas pertinentes as atividades do processo de software da Fujitsu são:

- Ferramenta para de projeto de software: Esta ferramenta utiliza em sua estrutura uma linguagem, que combina digramas e pseudo-código;
- Geradores de código: Ferramenta utilizada para automatizar a atividade de codificação. Trabalha a partir do pseudo-código gerado pela ferramenta de projeto e gera parte do código fonte;
- Ferramenta para automatizar a atividade de teste: Automatiza toda a atividade de teste desenvolvida pela fábrica de software em questão, simulando ambientes para execução de programas, analisando performance dos programas que devem ser executados em terminais compartilhados e, por fim, provê mecanismo para o armazenamento das informações geradas pela atividade de teste;
- Ferramenta para gerenciamento de documentos: Provê mecanismo para manipular os documentos na linguagem japonesa. Automatiza a edição

eletrônica de documentos. Traduz, automaticamente, documentos da língua japonesa para a língua inglesa;

- Ferramenta para automatização da atividade de manutenção: Provê mecanismo para o armazenamento das informações geradas na atividade em questão. Controla as solicitações de melhorias dos clientes;
- Ferramenta para controle de projeto: Provê mecanismo para o armazenamento e recuperação das informações dos projetos pertinentes à fábrica de software da Fujitsu e automatiza a atividade de gestão de projetos descrita, anteriormente.

Por fim, é necessário ressaltar que todas as fábricas japonesas possuem em sua estrutura de produção de software o conceito de biblioteca de componentes, caracterizando, assim, o conceito de reuso de software. A SDC também se encaixa neste segmento.

B.5 – Produtos Gerados e Forma de Criação e Organização do Processo

Esta seção tem como objetivo apresentar se as empresas japonesas e americana trabalham sob a ótica das orientações a domínio, a produto e a tecnologia. Uma outra vertente da seção se configura sobre a forma de criação e organização do processo.

Ao analisar a seção B.1 e com base nas considerações efetuadas por CUSUMANO (1991), é possível verificar que nenhuma delas trabalha orientada a domínio, produtos e tecnologia. Todas produzem software para vários domínios do conhecimento, utilizando várias tecnologias.

Já em relação à forma de organização do processo, as empresas japonesas e a SDC, levaram vários anos para se comporem como fábrica de software. Nestas

empresas, as iniciativas organizacionais relacionadas à questão fabril para o desenvolvimento de software iniciaram-se na década de 1970. Excluindo a Toshiba e a Fujitsu, a forma de criação e organização do processo das demais empresas foi, totalmente, baseada nos conhecimentos dos envolvidos com o processo de produção de software de tais empresas.

Na Toshiba além do conhecimento dos envolvidos, a criação do processo fabril foi inspirada nas boas práticas relacionadas à produção de hardware já consolidadas na empresa. O ciclo de produção de hardware em questão pode ser dividido em 3 grandes fases:

- Projeto de hardware: fase que engloba as atividades do projeto lógico e desenho do produto;
- Produção de hardware: traduz o conceito de manufatura a partir de componentes de hardware reutilizáveis;
- Teste: Os produtos manufaturados são testados para, posteriormente, serem colocados no mercado.

É possível estabelecer uma comparação entre ciclo de produção de hardware e o ciclo de produção de software, descrito na seção B.2.3. As fases de "projeto de hardware", "produção" e "teste" relacionam-se com as atividades de "projeto de software", "codificação" e "teste". Tal relação é verificada, nitidamente, na Figura B.6.

Figura B.6 – Relação entre o processo de produção de computadores (fábrica convencional) e o processo de produção de software (fábrica de software) da Toshiba (Adaptado de Cusumano (1991))

Ao analisar tal figura é possível perceber, também, que a atividade de projeto de software aglomera o desenvolvimento da arquitetura do software (analogia com o projeto lógico do hardware) e o projeto de programa (analogia com o desenho do produto). Já a atividade de codificação está dividida em programação com reuso (analogia com a montagem do projeto de hardware) e compilação, atividade responsável por checar se o código desenvolvido não possui erros (analogia com checagem do projeto de hardware). A atividade de teste de software é dividida em teste de módulo e teste de software, assemelha-se a atividade de teste de unidade e ao teste de hardware.

O símbolo representando a arquitetura do sistema apresentado na Figura B.6, representa uma solução integrada para um determinado projeto, envolvendo dispositivos de hardware e software.

Já em relação a Fujitsu, CUSUMANO (1991) ressalta que o processo de produção de software utilizou as seguintes estratégias em sua concepção:

- Desenvolvimento tecnológico: Busca por linguagens de alto nível que propiciassem uma alta produtividade no desenvolvimento do código fonte. Estas linguagens deveriam propiciar o conceito de reuso de código, através do desenvolvimento de funções e procedimentos (sub-rotinas);
- Padronização: Desenvolvimento de uma estrutura organizacional e um processo de produção de software padronizado;
- Mecanização e Automação: Aquisição ou desenvolvimento de ferramentas e sistemas computacionais que propiciassem um maior ganho de tempo na execução das atividades do processo.
- Suporte e Desenvolvimento Tecnológico dos Colaboradores: Desenvolvimento um programa de treinamento nas ferramentas e no processo padronizado.

É perceptível que as estratégias utilizadas pela Fujitsu são caracterizadas como genéricas e, também, devem ter sido utilizadas por todas as fábricas que compõe este trabalho.

O autor também deixa claro que as estratégias utilizadas pela Fujitsu foram de fundamental importância, pois elas nortearam todo o aspecto evolutivo do processo de produção de software implementado.